

Chapter 5

Label Generation for static Airport Charts

The previous chapters described the core data processing routines for labeling airport data. In these steps, AMDB data have been filtered, interpreted, and aggregated into new data structures more usable for label placement. These steps are also a general, application independent preparation for *decluttering* of redundant information.

The final steps for labeling airport maps are the generation and evaluation of label candidates and the final placement of labels (*deconfliction*). These last steps are specific for a target system and must be customized accordingly.

This chapter describes the label candidate generation, evaluation, and label placement for the labels used on static airport maps. Chapter 6 explains the same steps for an electronic Airport Moving Map application.

Furthermore, this chapter also outlines the generation of the airport map itself, so that the result will be an integrated solution for map and label generation.

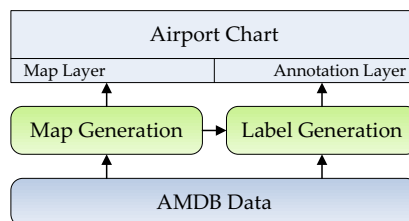


Figure 5.1: Overview of the system for map and label generation for airport charts.

The structure of this chapter is aligned with the mentioned steps. The generation of the background map is covered first, as some dependencies to the label candidate generation exist. After that, the candidate generation, evaluation, and the label placement are described. The chapter closes with the presentation of the resulting airport maps.

5.1 Map Generation

Similar to the label generation process, the map generation is based on three data sources: AMDBs for geo-spatial data, a symbol library for all kinds of symbology to be placed on the map, and finally a configuration database for all parameters defining a particular map.

5.1.1 Information Analysis

The main data source for the generation of airport maps are AMDB data. As the focus during the development of AMDBs has been on electronic airport displays, it had to be investigated, if all data required for the automatic generation of static airport maps are available respectively if and which additional data are needed.

An information and gap analysis has been performed to ensure the availability of all required data. This analysis included a comparison of the AMDB standards RTCA DO-272A and DO-291 with airport chart specifications [Doc8697, Jep04], as well as comparisons of existing AMDB data with existing airport charts.

115 information entities depicted on airport charts have been identified. Of these entities, about one third is available from AMDBs as is. This covers all large airport objects like runways, taxiways, or parking stands including their geometries and main attributes. Another third can be easily converted from AMDB data (e.g. different coding of runway surface materials), or loaded from other electronic aviation databases such as the Jeppesen Aviation Database or the Electronic Airport Directory.

The last third are information not available in electronic databases so far. These data entities are typically related to features which

1. are not visible on satellite imagery (the main data source for AMDBs), like all kinds of lighting systems,
2. are important for flight operations, such as take-off locations or runway incursion hotspots, but have no physical representation though they are tied to a specific geometric location, or
3. are located outside the airport boundary like roads, railroads, rivers etc. Such data can partially be taken from topography databases like VMAP, ATKIS, or from data providers like NavTeq and TeleAtlas.

Especially data from categories 1) and 2) are important information for pilots and a significant enhancement of existing aerodrome mapping databases.

After the identification of missing information, the AMDB database model has been updated accordingly, and the new data elements have been populated for a number of test airports. To cover a wide variety of different airport layouts and special situations, the airports of Denver (KDEN), Frankfurt (EDDF), Kansas City (KMCI), and London-Heathrow (EGLL) have been selected as test airports. The enhancements of the AMDB data model

and updates of the databases was joint work with Lisa Haskell and Bill Lugsch, Jeppesen, Denver.

Another important outcome of the analysis is, that some of the current charting rules are in the form “... chart when provided/specified/depicted by source ...”. Such rules cannot be implemented in an automatic chart generator as the original source (e.g. AIP) is not available to the system.

Legal reasons do exist in some cases like the mandatory display of the “Amendment number” for non-standard take-off minimums. In such cases, it is reasonable to capture the required information in the database.

In the majority of cases, it is more practical to define a standardized depiction, which is only driven by the data, and not by their representation in the AIP. This latter approach has been followed in this work. New business rules have been modeled, such that they are consistent with the existing rules and implementable with the available data.

5.1.2 Symbols

In general, a symbol is a graphical representation of an abstract database object. For example, a point feature with a single coordinate and type 'windsock' will be displayed using an according geometry.

As mentioned in the concepts chapter, two types of symbols have to be placed on paper charts. The first type is representing real world objects. In this case, the database has only textual or numeric information, e.g. about the type of approach light system (ALS) for a particular runway. To visualize this information, a geometry template matching the specific ALS has to be placed at the correct location and sized and rotated to match the real world situation.

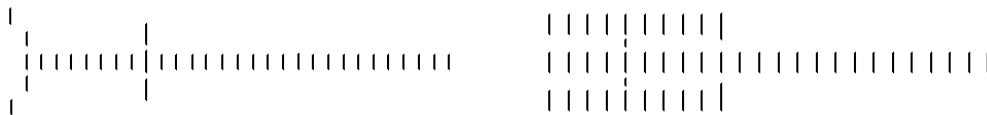


Figure 5.2: Symbols for approach light systems ALSF-1 and ALSF-2. (c) Jeppesen

The other type of symbols is representing airport objects which must be marked on a map, but in a more generic way. Such objects are, amongst others, antennas, windsocks, blast fences, or the aerodrome reference point. The layout and size of these pictograms can be defined arbitrarily as long as the user can identify them easily.



Figure 5.3: Symbols for aerodrome reference points, airport beacons, windsocks, light poles, antennas, and trees. (c) Jeppesen

All symbols have been stored in ESRI Shapefile format, such that they can be imported with their attached geometry. Additional information like outline or fill colors, opacity, and size (absolute or relative) are stored as attributes.

5.1.3 Configuration

A variety of different map types and layouts exist. In order to generate existing schemes and have the flexibility to configure and evaluate novel chart designs on the fly without modifying the source code of the application, all parameters have been externalized into configuration files.

An overall configuration exists to specify the elements of a complete chart: header, communication box, and airport map(s) and their location on the chart. This configuration refers to secondary templates specifying the content and layout of each chart element such as airport overview or parking stand maps. These templates are also specifying the mapping of symbols to base features, as well as fill and line styles for normal airport features in Styled Layer Descriptor (SLD) format [SLD].

The default configuration for an airport chart is set to contain exactly one airport overview map with a scale automatically adjusted to fit the complete airport onto the map. Without further parameters, only this setup is used.

To tailor this chart, an additional configuration file is read for each airport. In this file, all layout parameters for the chart or a particular map can be modified as needed. A frequent modification on large airports is the addition of one or more parking stand maps.

5.1.4 Implementation

The generation of the map itself is relatively easy with the preparatory work of the symbol library and the configuration database.

First, a private working copy of all GIS data is created for each map instance. Feature types not to be depicted on this map are excluded from the working copy. Each feature type is added as its own layer, so that it is still possible to work with the data in a GIS manner. The style for each layer is imported from the configuration database as described above.

After that, symbols like approach light systems are added to the map in their original size. ALS' are frequently extending the bounding box of the airport because of their length and their location at the end of runways. Thus, the envelope of the airport is computed not until the addition of this symbol category is completed. When the dimension of the bounding box is known, all feature (parts) outside can be clipped, and the remaining geometries can be scaled and simplified to the paper size and resolution of the map.

Before placing the remaining abstract symbols on the map, a decluttering of their base features is required. AMDBs are containing a vast number of vertical point and line objects which are in essence obstacles, but only very few of them are important enough to be depicted. The following rules have been applied to reduce their number to a degree

which can be depicted without cluttering:

- Place all symbols relevant to flight operations such as windsocks.
- Place all symbols with a critical height.

Currently this goal is implemented by suppressing all AMDB features which do not have a populated height attribute. Today, the height attribute is only populated if an obstacle is depicted with its height and elevation on the obstacle chart in the AIP. Only these obstacles are presented on manually generated paper charts and therefore it is reasonable to add only these to an automatically generated map.

When databases with all obstacles on and around an airport including the height of all obstacles will become available in the future, e.g. through LIDAR- or high resolution satellite radar-scanning (TerraSAR-X), this rule must be modified to display e.g. only those obstacles penetrating the obstacle assessment surface as specified in RTCA DO-276A [DO276A].

- If multiple features of the same type are located very close together, so that the respective symbols would overlap, the features are merged and only one symbol is placed in the center of the individual features. If required, the new symbol will be labeled with the elevation of the highest obstacle of the group.

The importance of the decluttering of obstacle symbols is demonstrated in figure 5.4. The map on the left shows all obstacles in the current database. Such a depiction is barely usable due to the large number of generic obstacle and tree symbols. In the map shown on the right side, the described rules have been applied. Only the important windsock is left, which is consistent with today's charts.



Figure 5.4: Decluttering of symbols, shown on an example from Toulouse-Blagnac.

5.1.5 Results

The maps in figure 5.5 are depicting the assembly of an airport map graphically. On the first map, only the normal AMDB features are drawn in the typical style for airport overview maps. On the second map the fixed size symbols like approach light systems are added. After that, all other symbols are placed.

The last map includes additional cultural features like highways, railroads, power lines, and rivers in the vicinity of the airport. Today's airport charts are showing these features to a varying extent. Most of them can be regarded as distracting and are not displayed in

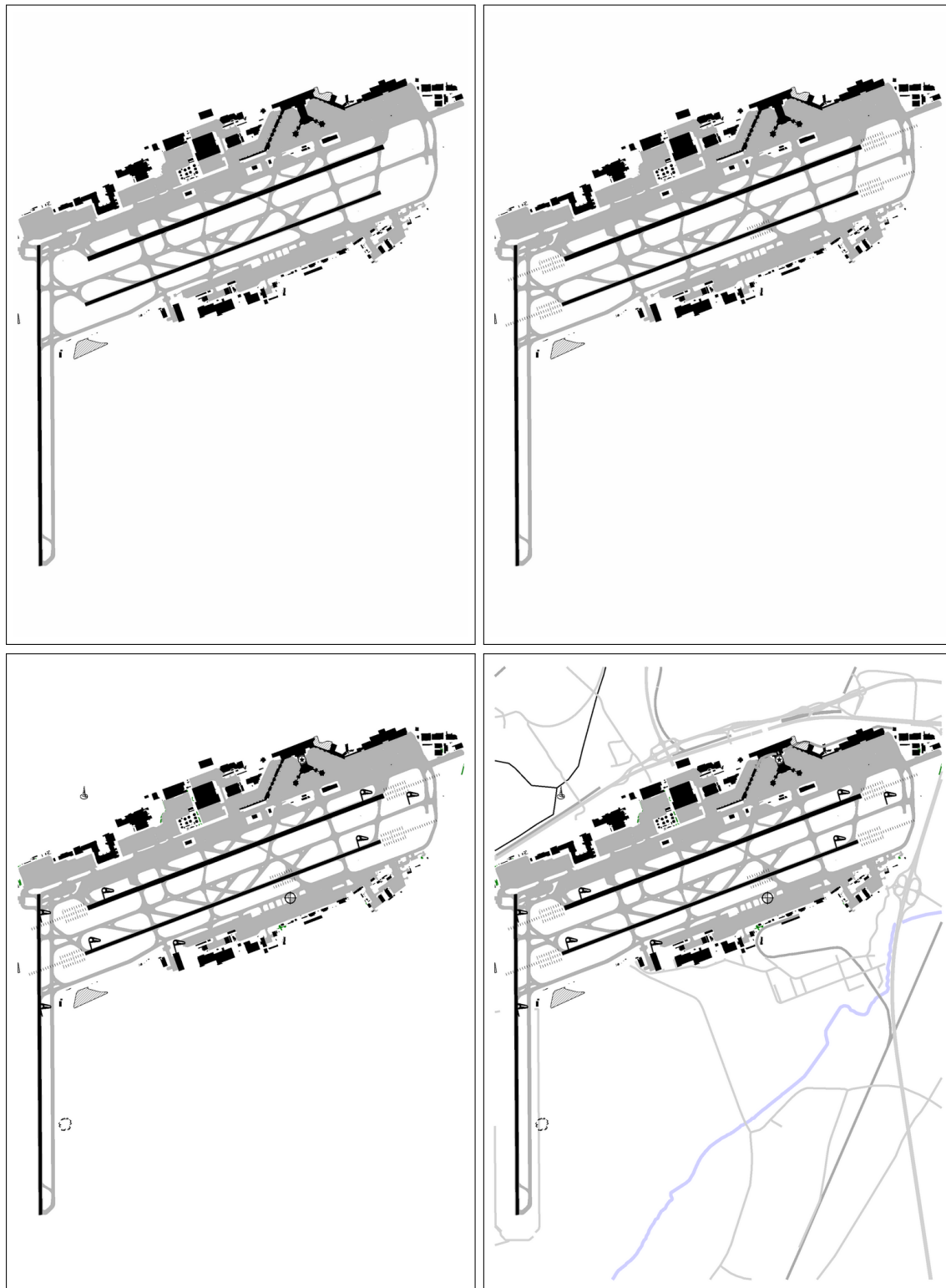


Figure 5.5: Assembly of an airport map. Starting with only AMDB features (top left), fixed size (top right) and resizable symbols (bottom left) are added, as well as cultural features in the vicinity of the airport.

this work. Only streets and highways are shown, as those are occasionally used for visual reference by pilots during the final approach.

The implementation and manual validation of airport maps showed, that all data required for a complete map are available in the upgraded AMDB databases. The architecture of the mapping component supports a free configuration of the map content, including styles of AMDB features and mapping of AMDB objects to arbitrary symbols. The maps generated by the described system are close to original charts. Deviations are frequently caused by varying depictions on existing paper charts, caused by the artistic license of chart compilers.

5.2 Generation and Evaluation of Label Candidates

After the generation of the background map, the labeling process is performed. This involves the steps of generating label candidates and computing a labeling solution with the maximum number of labels placed.

In this chapter, the generation and evaluation of label candidates will be outlined. A label candidate is a possible location for the final label position. A label candidate must be located such that it can be easily and unambiguously associated with its base feature.

To identify label candidate locations, the specification of today's airport charts [Jep04] has been analyzed with regards to the airport elements to be labeled and the exact style they have to be labeled. This document combines the regulatory requirements for charts [ICAO4, Doc8697] with existing cartographic knowledge. Human factors aspects are already integrated in this specification. Following the mentioned rules therefore guarantees a good legibility and unambiguity of the resulting labeling solution.

A sample of existing paper charts has also been analyzed to study the practical application of the specification and in order to identify solutions for special situations which are not covered by the procedures manual.

Based on the analysis, computer implementable algorithms have been defined to compute locations where labels can be placed. Besides the good and unambiguous association between the returned locations and the feature to be labeled, it is also important from an algorithmic standpoint to produce a large variety of different candidate locations. This helps during the deconfliction process to find conflict-free candidates for label placement, and to find a labeling solution with maximal size.

The other important step is the evaluation of label candidate positions. A single quality factor must be defined for each candidate location considering general cartographic factors (as mentioned in section 2.1.4) as well as factors specific for airport charts. This quality factor will be used during the deconfliction to optimize the chosen solution by maximizing the total quality of all selected candidates.

The quality value computed here considers only static factors like overlaps with map elements. Dynamic components, such as overlap with other label candidates, are computed temporarily during the label placement phase.

Both the generation of label candidate positions as well as their evaluation have been implemented in an expert system utilizing the data structures defined and assembled in chapter 4.

Depending on the geometry type of the source feature, common point-, line-, or polygon-candidate models will be used for the generation of label candidates

The value range of the quality factor has been defined from 1000 (optimum quality) to 0 (not suitable for placing a label). The quality values for different label types are not necessarily comparable, but they must be suitable to identify a ranking of the best candidates for a particular map element among all of its associated candidates.

Depending on the size and complexity of the airport, different sets of maps are available. For small airports, only the general airport overview chart exists. In this case, all map elements must be labeled on this chart.

Additional “parking stand charts” with higher scales exist for larger airports. The maximum level-of-detail must be shown in the zoom-ins, while certain information can be generalized or completely suppressed in the overview chart.

Different label types used for these different map types are also described in the following sections.

5.2.1 Runways

Runways are the most critical elements of airports and must be labeled under all circumstances on charts. This section lists the specifications and implementation accordingly of all labels to be placed in conjunction with runways.

5.2.1.1 Runway Designator

5.2.1.1.1 Specification

RWY01: A label with the runway designator and the magnetic runway bearing shall be placed at the end of the respective runway direction. The two text fields are embedded in an oval outline as shown in figure 5.6. The symbol is rotated parallel or perpendicular to the runway direction, whichever is closer to the upright position.

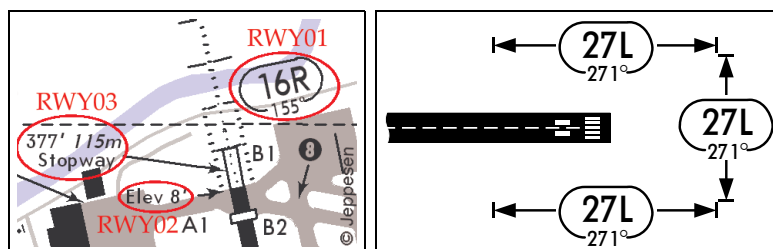


Figure 5.6: Placement of the Runway Designator.

The runway designator or runway number label is logically associated with the runway threshold in AMDBs. Frequently, an approach light system (ALS) is connected to the

threshold, thus extending the area where this label can be placed while still maintaining a good correlation with the runway. Accordingly, label candidates can be placed using a slider model as depicted on the right side of figure 5.6.

5.2.1.1.2 Implementation The generation of such label candidates has been implemented using a guideline representing the centers of all possible candidates. This line is computed by buffering all runway geometries (including the ALS) by the half height of the new label. The guideline is cut on the runway side on the level of the thresholds plus the half width of the label. Resulting guidelines for runways “25L” and “25R” in Frankfurt/Main are illustrated as red lines in figure 5.7.

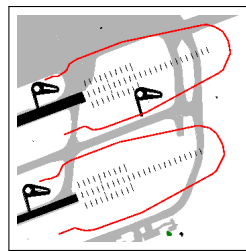


Figure 5.7: Guidelines for runway designator labels “25L” and “25R” in Frankfurt/Main.

After that, label candidates are placed along the guideline. Per definition, candidates can be placed anywhere on the guideline. An approximation has been used here to place a candidate every 50 meters. The following evaluation of the candidates is done by setting the quality factor for each candidate to the maximum value of 1000 first and subtracting penalties when one or more of the following rules are met:

- The distance between the threshold and the label candidate times 0.5 is subtracted from the quality factor to fulfill the requirement that the label should be placed close to the runway end. As the longest ALS is about 800 meters long, this can contribute to a penalty of up to 40 points.
- To avoid placement of labels *between* two parallel runways, which would be highly ambiguous, according candidates must be degraded. A proximity factor of $(\text{symbol size} / \text{distance to other runway})^3$ has been found to have a good selectivity to identify *close* runways, while a heading factor of $e^{(\Delta(\text{heading})/15)^2}$ is used to verify if both runways are also nearly *parallel*. The parameters in the heading factor are adjusted, such that the factor is 1.0 for true parallel runways, and ≈ 0.5 for runways with different bearings of about 10 degrees.

The rational behind the heading factor is that runway designators are rotated. An ambiguous situation only occurs if the label is close to a second runway with the same heading. If the heading is different, a unique association exists by the orientation of labels and runway.

Thus, only the combination of the distance and the heading factors indicates a critical situation. In such cases, another geometrical check is performed to avoid an

unnecessary derating of unambiguous candidates. A direct line from the label candidate to the closest point of the other runway is constructed. If the candidate is separated from the other runway by the own runway, then the situation is unambiguous and the candidate can be kept. If the line does not intersect its own runway, then the label candidate is located between the own and the other runway. This candidate is derated because it cannot be clearly associated with the correct runway end.

- The quality of candidates overlapping with a runway element, a runway intersection, a stopway, a clearway, or an ALS of another runway is set to 0. This is done, because the placement of the label at the location of this candidate would be highly ambiguous, too.
- If a candidate overlaps with other map features like apron or taxiway elements, a penalty depending on the fraction overlap area/label area and the contrast between the label color and the color of the other feature is subtracted. This rule is supposed to optimize the legibility of the label by preferring candidates with no conflicts at all or conflicts with a high contrast.
- Candidates being located at least partially outside of the defined map area are flagged as unusable by a quality factor of 0.

Label Candidates with a quality of less than or equal to 0 are unusable and are excluded from further processing.

5.2.1.1.3 Results Figure 5.8 shows three examples for the label candidate generation and evaluation. The outline of the label candidates are symbolized by the amber geometries with inscribed quality factors. In these examples, label candidates are spaced at 200 meters for better legibility of the figures.

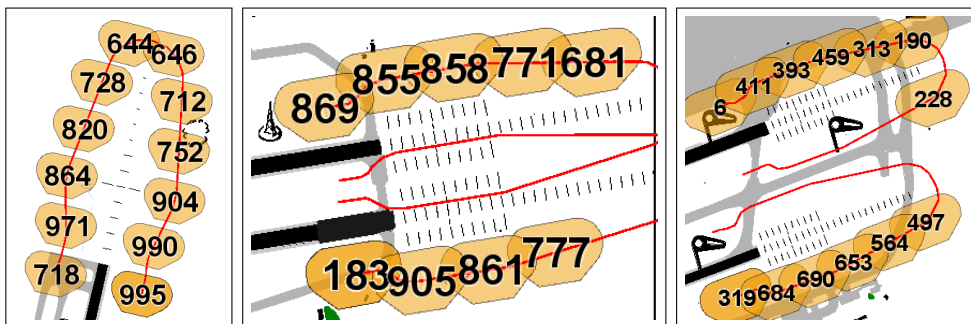


Figure 5.8: Candidates for runway designator labels.

The left image shows runway number candidates for a single runway in Kansas City International. The label candidate in the lower left corner has a lower quality factor due to the overlap with the taxiways. Similarly, two candidates on the upper right side are degraded due to the overlap with the tree symbol. The quality values of the other labels are dominated by the increasing distance from the threshold (bottom to top).

The center image shows label candidates for two very close parallel runways in Berlin-Tegel. This example demonstrates that all label candidates *between* the two runways are removed reliably due to the risk of ambiguity. The labels on the detached sides of the runway are affected only to a small extent.

The right picture of figure 5.8 shows a complex situation with overlapping taxiways and symbols in Frankfurt/Main. One label candidate of the north runway is kept at the side pointing towards the parallel runway, but due to the larger distance between the two runways and the additional offset of the thresholds, this situation is tolerable. Even in this case, the quality factors for the label candidates are set reasonably.

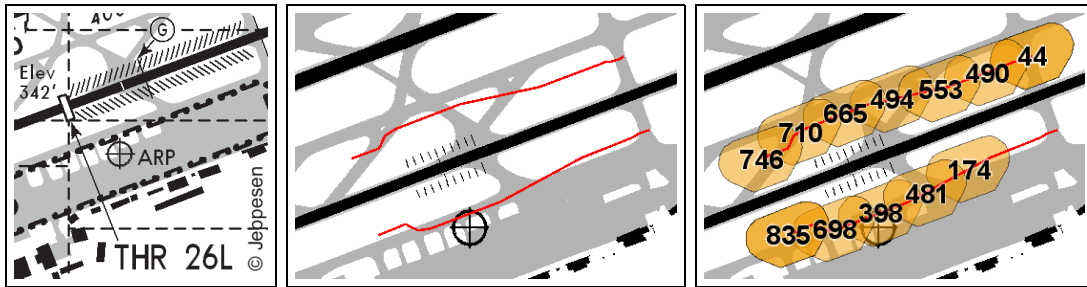


Figure 5.9: Runway designator label for the HALS/DTOP threshold “26L” in Frankfurt/Main.

An interesting situation is again the HALS/DTOP threshold “26L” in Frankfurt/Main (figure 5.9). This situation is not covered by the existing specification which ties the runway designator labels to the runway ends. In consequence, this situation is depicted in a non-standard way as displaced threshold on today’s chart, and labeled using a simple text field without magnetic bearing or oval outline. The lighting system and the elevation label are placed as for normal runways.

In the AMDB of Frankfurt, threshold “26L” is modeled as a normal threshold feature (not displaced threshold). Thus, it will be handled like any other runway threshold by the system. Guidelines and label candidates are generated and evaluated using the process described above. The approach light system is drawn as part of the map layer. A default symbol is used as the non-standard HALS/DTOP ALS is not available in the symbol library.

This handling is different than the current depiction on paper charts, but it offers a higher consistency with other runway number labels.

5.2.1.2 Runway Elevation

5.2.1.2.1 Specification

RWY02: The runway end elevation shall be depicted in feet.

Usually the string “Elev” is placed in the first line, and the actual elevation in a second line. Only in rare cases, the label is formatted using a single line (see e.g. figure 5.6 on

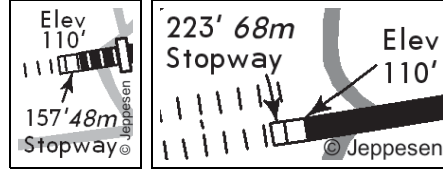


Figure 5.10: Placement of the Runway Elevation and Stopway Labels.

page 124). For consistency, only the two-line version will be used in this work.

The runway end elevation attribute is connected to the runway threshold in the database. Due to the limited amount of available space in this area, the elevation label is usually placed at some distance and connected with the runway end by an arrow (see figure 5.10).

5.2.1.2.2 Implementation The desired location(s) of runway elevation labels are not specified exactly. Hence, a generic polygon candidate model is used to handle this situation in a flexible way. A polygon candidate model works similar as the guideline model, but with the difference that the guidance geometry is a polygon defining possible label locations. In general, label candidates can be placed anywhere within the defined geometry. In this implementation, an approximation using discrete points is used. The points are placed on a raster defined by the bounding box of the guidance polygon and are spaced, such that they fit exactly into the envelope. A label candidate is placed at each point.

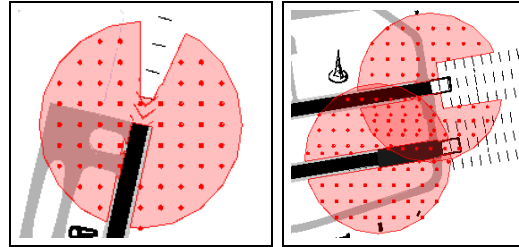


Figure 5.11: Guidance polygons for runway elevation labels.

For runway elevation labels, a circle with a radius of 250 meters around the threshold coordinate has been defined as guidance polygon. The runway geometry is subtracted from this circle, as the label must not be placed there. Label candidates are placed within the circle on a raster of approximately 50 meters.

Figure 5.11 shows the resulting guidance polygons as red circles, and the candidate reference points as small red dots. The right image shows the two parallel runways in Berlin-Tegel. On the south runway, the elevation attribute is attached to a displaced threshold instead of the standard threshold. Therefore, the guidance geometry is not attached to the runway end, but to the location of the displaced threshold.

The evaluation of the label candidates is similar to the evaluation of the runway designator labels. Starting with a quality factor of 1000, points are subtracted if the candidate has particular deficiencies, identified by the following rules:

- The distance between the threshold and the label candidate, normalized by the radius of the guidance geometry, is subtracted from the quality factor, such that a maximum penalty of 350 is assigned. This rule targets at locating the label close to the runway end and thus at a good assignability of the label with the runway.
- To avoid placement of labels *between* two parallel runways, which would be highly ambiguous, according candidates must be degraded. A proximity factor of $(\text{symbol size} / \text{distance to other threshold})^2$ has been found to have a good selectivity to identify *close* runways.

A further distinction by the heading of the involved thresholds as used for the runway designator labels is not applicable, as elevation labels are not rotated.

Nevertheless, the geometrical check can be performed to evaluate if the candidate is separated from the other threshold by the own runway geometry. If this is the case, the situation is unambiguous and the candidate can be kept. Otherwise the label candidate is derated because it cannot be clearly associated with the correct runway end.

- The quality of candidates overlapping another runway geometry (including ALS) is set to 0. This is done, because the placement of the label at the location of this candidate would be highly ambiguous, too.
- The quality of candidates overlapping other map features is derated using the same algorithm as described for runway designator labels.
- Candidates being located at least partially outside of the defined map area are flagged as unusable by a quality factor of 0.

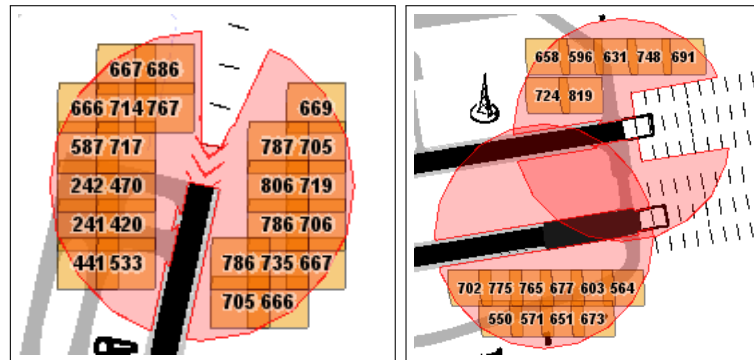


Figure 5.12: Candidates for runway elevation labels.

5.2.1.2.3 Results Figure 5.12 shows two examples of runway elevation candidates, represented by the amber boxes. The inscribed numbers are the assigned quality factors. To improve the legibility of the screenshots, the distance between the candidates has been increased to 100 meters in both examples.

The left image shows a simple situation in Kansas City with little interference with other map elements. The quality values of candidates located to the east of the runway are only affected by their distance to the threshold. The values to the southwest are also influenced

by overlaps with the taxiways. The best candidate defined by these rules would be direct to the east of the threshold with a quality value of 806.

The right illustration is again from Berlin-Tegel. All label candidates between the runways are removed reliably. The best candidate for the label on the north side is situated in the free space to the northwest of the threshold. The best candidate for the south runway is located to the southwest, slightly avoiding an overlap with the taxiway. Both placements are reasonable and comparable with the manual placement on existing paper charts.

5.2.1.3 Stopway Length

5.2.1.3.1 Specification

RWY03: If the runway has a designated stopway, the length of the stopway shall be charted in feet and meters together with “Stopway”.

This label can be located on both sides of a stopway. In case of lacking space, the label is placed remotely and connected by an arrow similar to the elevation label (see figure 5.10 on page 128). All other properties of this label are similar to the runway elevation label as well.

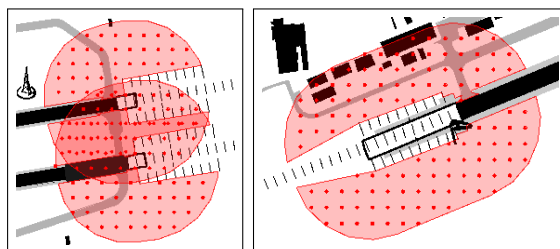


Figure 5.13: Guidance polygons for stopway length labels.

5.2.1.3.2 Implementation Candidates for the stopway length label are modeled by using a polygon model. A buffer of 250 meters around the stopway feature, minus the outline of the runway is used as guidance polygon. Within this geometry, label candidates are placed at distances of 50 meters.

The guidance polygon and the candidate centers are illustrated in figure 5.13 for Berlin-Tegel and Manching. The stopways in Berlin are relatively short. Thus, the resulting guidance geometry is nearly circular. The guidance polygon for the longer stopways in Manching are extended to a long oval, thus leading to a larger number of candidates and a better chance to find a good, deconflicted labeling solution.

The evaluation of these label candidates is done identically to the candidates for the runway elevation.

- The distance between the candidate and the stopway is the criteria to define the correlation of the label with its feature.
- Ambiguous placement too close to other stopways is prohibited.

- Candidates overlapping other map content are derated due to bad legibility.
- Candidates overlapping runways or being located outside of the map area are removed completely.

Details can be found in section 5.2.1.2.2.

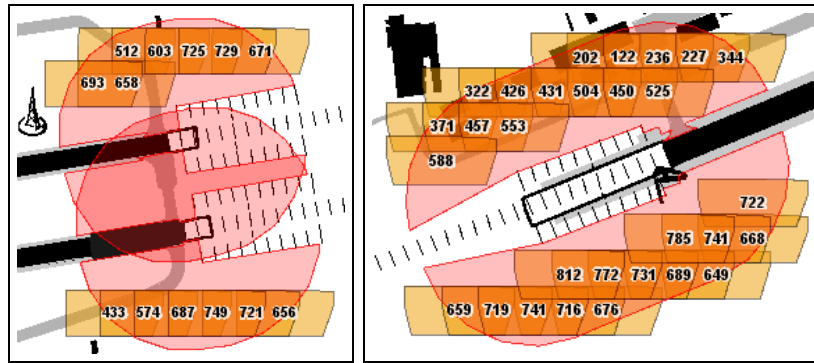


Figure 5.14: Candidates for stopway length labels.

5.2.1.3.3 Results Figure 5.14 shows two examples for evaluated label candidates. The quality values of candidates not overlapping any map features are controlled by their distance to the stopbar. Candidates overlapping map elements or being close to other runways are degraded accordingly. In both cases, the label candidate with the highest score is located close to the stopway and clear of any map elements.

5.2.1.4 Runway Length

5.2.1.4.1 Specification

RWY04: A label for the runway length shall be placed alongside the runway and rotated parallel to the runway. The length shall be listed in feet and meters.

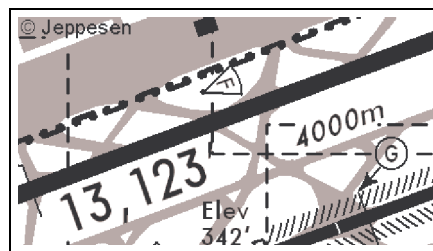


Figure 5.15: Placement of the runway length label.

5.2.1.4.2 Implementation The implementation of the runway length label is done similarly to the runway designator symbol. First, two guidelines are generated to the left and to the right side of the runway by using all elements of a particular runway container.

Along these lines, label candidates are generated at a distance of 50 meters between each other.

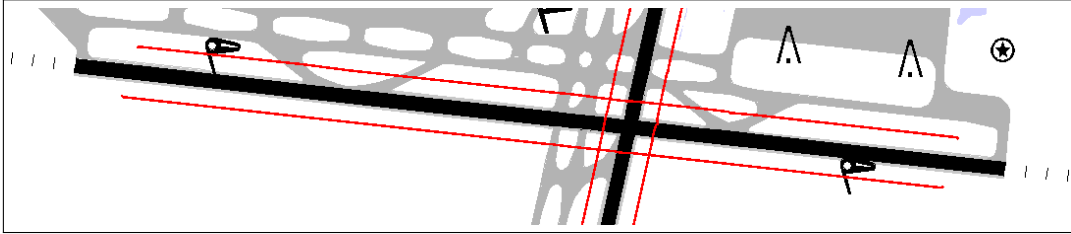


Figure 5.16: Guidelines for runway length labels in Kansas City.

The rules used to evaluate the candidates are also adapted from the runway designator algorithm. Starting with a quality factor of 1000, penalties are assigned for a number of factors.

- The distance from the runway center normalized by the total runway length is subtracted from the quality. Thereby, a maximum penalty of 500 can be assigned. This rule is used to prefer locations near the runway center. This is done for faster information retrieval, as this is the first spot where pilots look for this information.
- Label candidates located between close, parallel runways are derated to avoid an ambiguous placement at this location. The same rules are used as already described in section 5.2.1.1.2.
- Candidates overlapping other runways or not completely within the map area are flagged as unusable.
- Candidates overlapping other airport features like taxiways or aprons are derated depending on the intersection area and the type of the other feature. This is an important rule, as conflicts with runway exits are nearly inevitable due to the location and size of the runway length label.

5.2.1.4.3 Results Figure 5.17 shows the outline of the candidates (amber boxes) and assigned quality factors for the runway length labels of runway “9/27” in Kansas City and both runways in Berlin-Tegel. For a better visibility, the candidates are placed at a distance of 200 meters instead of the normal 50 meters.

The first situation illustrates the effect of the overlap mechanisms. No label candidates are placed in the area of the intersecting runway. Furthermore, all candidates on the north side of the runway are lower rated than the label on the south side due to the overlapping runway exits. The decreasing quality factors from the runway center to the runway edges are also clearly visible.

The second example shows the parallel runways of Berlin-Tegel. Label candidates located between the two runways are removed reliably to avoid ambiguous label placement.

The results of the label candidate generation and evaluation for runway length labels have been manually validated for a number of airports, and the results are very promising for

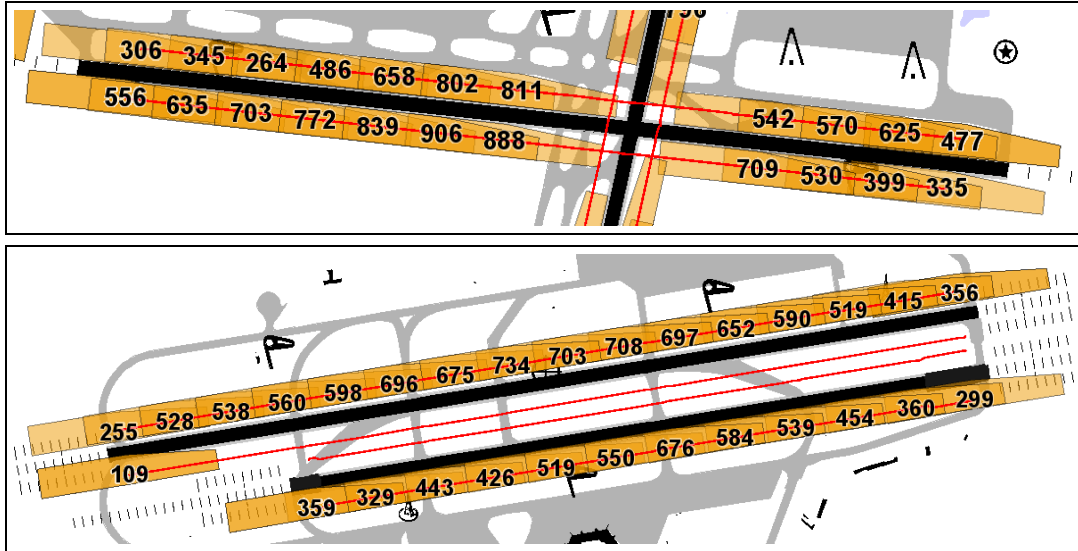


Figure 5.17: Candidates for runway length labels.

the next step of label placement.

5.2.1.5 Runway Designator for parking view maps

5.2.1.5.1 Specification

RWY05: The runway designator for both runway directions shall be depicted in the format “RWY “ + first direction + ‘/’ + second direction (e.g. “RWY 07L/25R”). This label shall only be placed on ground movement or parking view charts. All other runway labels shall be omitted in this case.

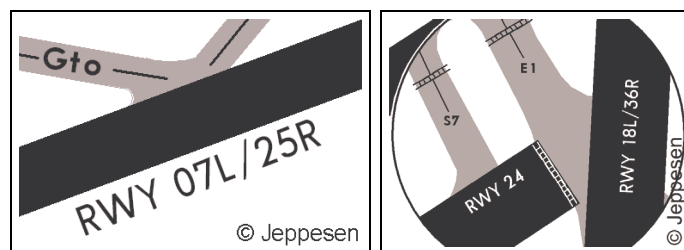


Figure 5.18: Placement of the runway designator label on parking view maps.

The label is placed near the center of the visible part of the runway and rotated parallel to the runway axis. In most cases, it is shifted to the left or the right side (left image in figure 5.18), but in some cases it is also centered within the runway boundary (right image).

For consistency reasons, it has been decided to use only the second placement of the label within the runway boundary. This solution offers the advantage of utilizing unused space

within the runway boundary. This improved the performance of the labeling solution for labels of this type as well as for other labels in the direct vicinity of the runway. For legibility, the label must be printed in white color on the black runway.

5.2.1.5.2 Implementation As in other cases, the implementation of this label type is accomplished by generating a guideline for label candidates first. This guideline is computed as the polygon medial axis (“skeleton”) [Lee82, Aic95] of all runway element, intersection, and displaced area geometries of the particular runway container intersecting the map area. The advantage of this implementation over the use of the painted centerline feature is that the skeleton can be derived from all polygons, while the painted centerline does not exist on runway displaced areas.

The derived line is shortened at the ends by the half label width to avoid misplaced labels.

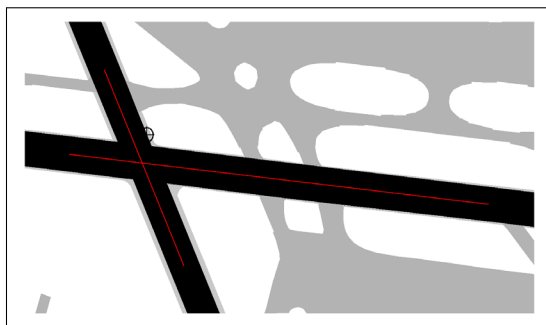


Figure 5.19: Guideline for runway designator labels.

Label candidates are generated along the guideline at a distance of 50 meters, or at 1/20th of the map extent, whichever is smaller. Both parameters are chosen, such that a fair number of candidates are generated. On one side, a large number is helpful to identify the best label position, while on the other side a smaller number is preferable for a faster runtime of the algorithm. The 1/20th rule is necessary here, because these labels are frequently placed on maps with large scales. In such situations, 50 meters between two candidates would not be a good approximation any more.

The rules for the evaluation of runway designator candidates are simple compared to the previous label types.

- The quality of label candidates overlapping another (intersecting) runway is reduced by 300 points. By doing so, the probability is low that such candidates are used for the final placement, but it is still possible if no other candidates do exist. Placing a label on the intersection of two runways is not ambiguous, as the runway headings must be different.
- The quality of label candidates is reduced by the distance from the (visible) runway center linearly, such that a candidate at the runway end has a quality factor of 0.

5.2.1.5.3 Results The resulting label candidates are shown in figure 5.20 on an example of two intersecting runways in Zürich-Kloten.

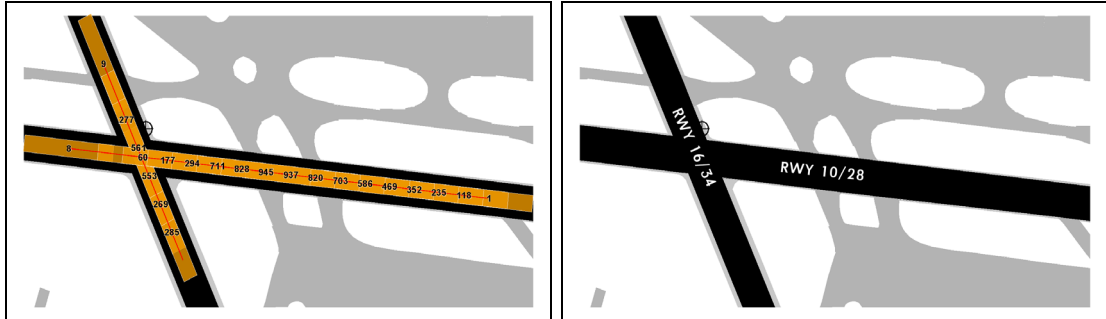


Figure 5.20: Candidates for runway designator labels.

The quality values of the candidates on runway “10/28” (east/west) are primarily influenced by their distance to the runway center. Only five candidates on the western end are overlapping with the intersecting runway and are degraded accordingly, one of them to a degree, such that it is not shown any more.

All candidates of the intersecting runway “16/34” besides the two at the ends are overlapping with the other runway. Therefore, all are degraded. Nevertheless, the candidate in the center has still the highest quality.

Using the candidates with the highest quality factors for label placement, results in the depiction on the right side of fig. 5.20. It is reasonable to assume, that the final label placement will look identical as no other label type candidates can interfere with the runway designator candidates.

5.2.2 Helipads

Helipads have the same importance to helicopters as runways do for fixed wing aircraft. Thus, they must be depicted and labeled with a very high priority on airport maps.

HELI01: Helipads not having an explicit designator are depicted by an “H”, enclosed in a triangle. This symbol is to be placed in all views.

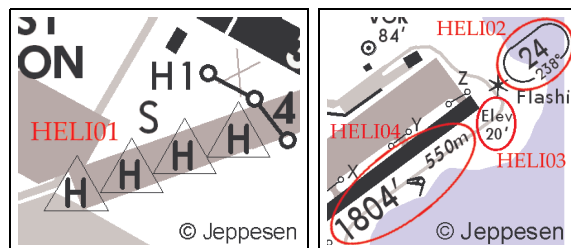


Figure 5.21: Placement of Helipad and Helicopter Runway labels.

This is the standard depiction of a helipad. The symbol must be placed at the coordinates of the helipad threshold. Thus, a deconfliction is not necessary and not possible.

A comparison of available helipad data in AMDBs with paper charts reveals some discrepancies. On a number of airports, one or more helipads are listed in AMDBs, but not charted on paper charts. As there are no evident reasons for this situation and for the sake of a consistent map, all available helipads will be charted in this work.

Some rare airports like Sumburgh, UK do have helicopter runways. The depiction of these helicopter runways is identical to conventional runways. It is just noted in the additional runway information section of an airport chart that a particular runway is for helicopter use only.

The specification of other helicopter runway labels is in accordance with regular runway labels.

HELI02: In the airport overview, a label with the helipad designator and the magnetic runway heading is placed at the beginning of the respective runway direction, rotated parallel or perpendicular to the runway direction whichever is closer to the upright position.

HELI03: The runway end elevation must be depicted.

HELI04: An additional label with the runway length and width in meters and feet is placed near the center of the runway shifted to the left or the right.

The rules for label candidate locations and candidate evaluation can be transferred from sections 5.2.1.1 - 5.2.1.4. Therefore, and due to the few occurrences, the labeling of helipad runways has not been implemented.

5.2.3 Buildings

5.2.3.1 Specification

Different types of buildings exist on an airport. Terminal buildings and hangars come into mind first and are, in fact, important because they are start and end points for common taxi-in or taxi-out maneuvers. A second category of buildings having “operational significance” [ICAO4] is even more important. These are buildings like the Control Tower, “Aircraft Rescue and Fire Fighting” services (ARFF) or the “Aviation Information Services and Aeronautical Meteorological Office” (AIS + MET). They must be depicted and labeled, even if they are very small compared to other buildings.

Fixed requirements about the placement of building labels do not exist. A variety of locations is used today, ranging from labels placed next to a building over to remote placement using an arrow connector to placement within the building (see fig. 5.22).

Building labels can be hyphenated.

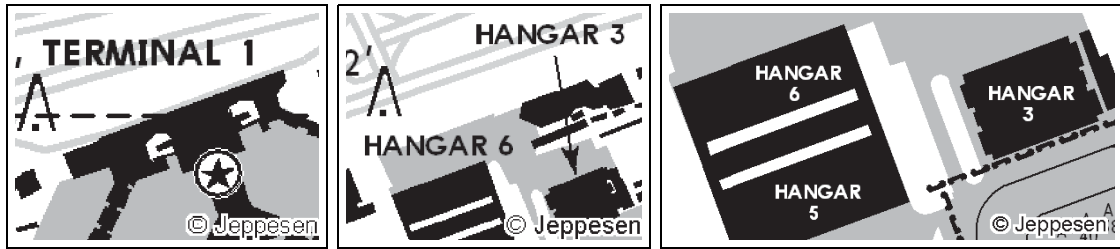


Figure 5.22: Placement of building labels on airport maps.

5.2.3.2 Implementation

A flexible solution to identify open space for label candidates in an area around a feature is the polygon candidate model introduced in section 5.2.1.2.2. A guidance geometry in the form of a buffer around the feature is defined. Label candidates (defined by their center) are allowed to be placed anywhere within the guidance geometry. In practice, this schema is approximated by placing the label candidates on a discrete raster.

Two different guidance geometries are calculated for label candidates to be placed within a building and on the outside. For label candidates around the building, the building outline is buffered by a distance of 250 meters. The width of a label candidate by two is added to the width in x-direction and the height by two in y-direction. This guarantees that the maximum distance between the label candidate outline and the building outline matches the 250 meter value.

Label candidates may not intersect the outline of a building, as labels must be located either completely within or completely outside a building. To avoid the generation of unnecessary label candidates at such locations, an according buffer area around all buildings is removed from the guidance geometry (see left image in figure 5.23).

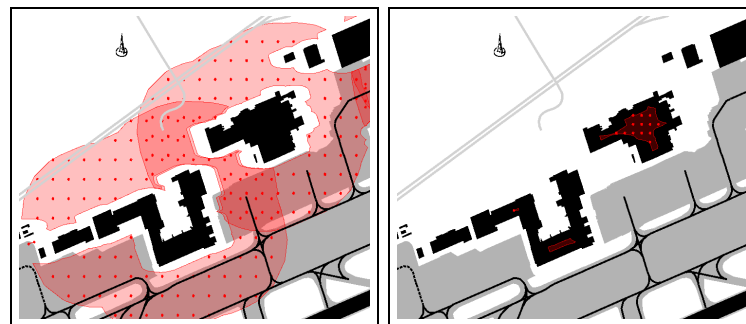


Figure 5.23: Guidance polygons for exterior (left) and interior (right) building labels.

The guidance geometry for interior label candidates is similarly computed by buffering the building outline by the label height/width to the inside of the building. This is illustrated in the right image in figure 5.23. The resulting guidance geometries (red shaded areas) are very small in both buildings and may be even non-existent on small map scales with

large label dimensions. If existing, label candidates within buildings are preferred for label placement. They can be associated clearly with their map element, and they do not contribute to label cluttering as no other labels can intersect a building.

As before, label candidates are placed on a 50-meter raster within the exterior part of the guidance polygon. The interior polygon candidates are placed at only half distance. This increases the chance to find valid candidate locations without overlaps of the building outline. The larger number of such candidates is unproblematic as it is limited by the small area of the interior guidance polygon.

If the name of a building contains a whitespace or a dash, two candidate types will be generated for each location simultaneously. Normal, single-line candidates, and word wrapped multi-line candidates. The higher quality value will decide which one to use during label placement.

As the first, common step during the evaluation of building label candidates, all candidates being located at least partially outside the map area or intersecting the outline of the building are rated with a quality factor of 0 due to bad legibility, and are thus effectively removed from the candidate set.

The remaining candidates are evaluated differently, depending on their location within or outside the referenced building. Interior candidates are only rated based on the distance between the center of the building and the center of the label candidate. Starting with the default value of 1000, a maximum value of 100 points is subtracted, such that a quality value of 900 is remaining for a candidates located at the outline of the building. Very high quality factors are assigned by doing so, supporting the idea that interior labels should be favored.

Exterior candidates are processed by a more complex rule set:

- Overlaps with map elements reduce the quality of a label candidate due to less contrast, impaired legibility, and possible ambiguities. The implementation considers the area of overlap and the type of the conflicting map element.
- The distance of a candidate to the labeled map element is also weighed, such that a maximum penalty of 500 points is assigned for labels at the maximum distance of 250 meters. To improve the unambiguity of remote labels, a connector arrow similar to the one used for runway elevation and stopway labels is used, if the distance is greater than 100 meters.
- To increase the unambiguity of candidates further, the distance of a candidate to other buildings is also calculated. A penalty up to 200 points is subtracted from the quality factor if a label is closer to another building than to the own building.

Applying these rules to all label candidates results in the desired candidate quality.

5.2.3.3 Results

A typical example for generated and evaluated building label candidates is shown in figure 5.24. For better legibility of the inscribed quality factors, the candidates in these screen-

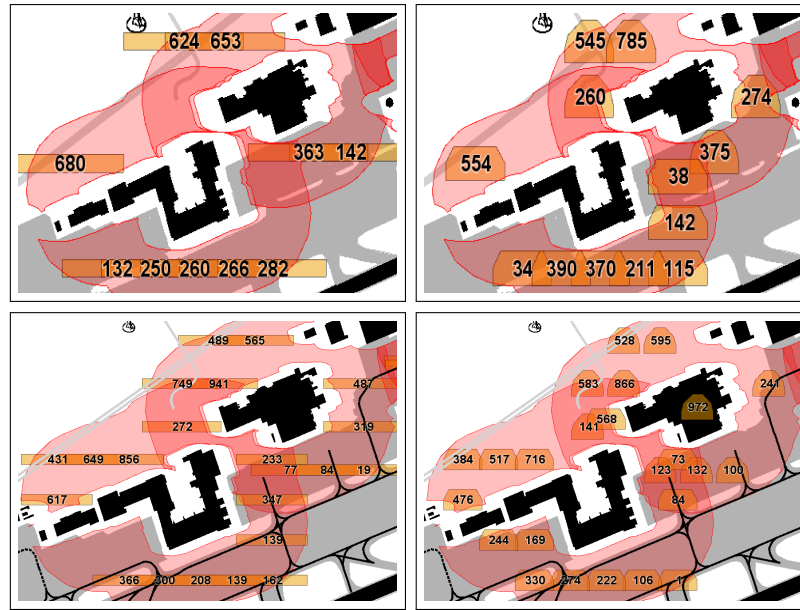


Figure 5.24: Evaluated label candidates for buildings.

shots are placed at a distance of 200 meters instead of the usual 50 meters. Candidates with a quality of less than or equal to zero are deemed unusable and are removed.

The images on the left side show single-line candidates (amber boxes) for the “Cargo Building” (left) and the “LH Cargo Center” (right). Both label strings can be hyphenated. The resulting multi-line candidates are shown in the two pictures on the right. The guidance geometries are illustrated as red areas.

The two images in the top row are extracted from a small scale airport overview chart. Thus, the labels are relatively large compared to the building.

The two pictures in the bottom row show the same content, but taken from a large scale map. Due to the smaller label candidates, the restricted areas around the building outlines are smaller. In consequence, more label candidates can be placed, and enough space becomes available for one multi-line label inside the “LH Cargo Center”. Chances are high that this interior candidate is used for label placement as it is rated with the best quality factor among all candidates of this feature, and overlaps with other candidates are unlikely.

The quality factors of the exterior candidates are representing the described rules. Label candidates located close to both buildings are rated worse than other labels of the same map element due to the ambiguity of these locations. Quality values are also decreasing with increasing distance from the building.

Another observation is that multi-line labels tend to be rated higher when placed at the same location as single-line labels. This can be attributed to the fact that they are more compact and thus more remote from other buildings even if the distance to the own building is the same. Thus, it can be expected that statistically slightly more multi-line than single-line labels are used for final label placement. This also contributes to a more

unambiguous labeling solution.

Imaging that the label placement algorithm will try to place the label at the candidates with the highest quality per feature will lead to good solutions in all cases.

5.2.4 Taxiway Labels

5.2.4.1 Specification

The official specification for taxiway labels is rather short:

TAXI01: Taxiway labels should be placed *next* to the appropriate taxiway. The label should only be placed *within* the taxiway if unable to clearly identify the taxiway otherwise.

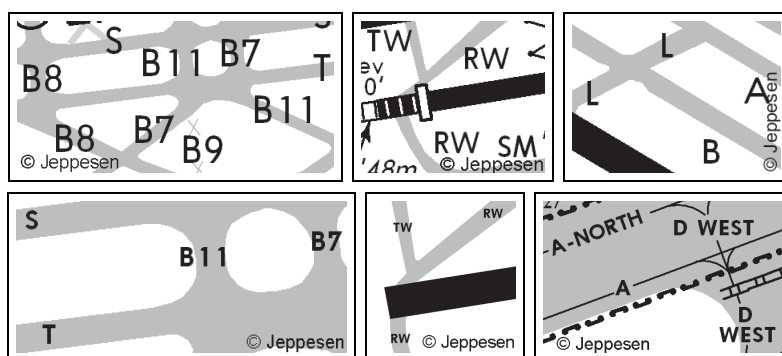


Figure 5.25: Placement of taxiway labels on airport overview (top row) and parking view maps (bottom row).

An analysis of existing charts (fig. 5.25) provides a better insight into the labeling schema used on today’s charts. Both placements near and within the taxiway geometry are used on airport overview charts. On parking charts, the labels are prevalingly placed on the taxiway. Criteria used to decide on the placement concept are non evident. Thus, the current label placement must be attributed to the artistic license of the chart compilers. An additional characteristic, which can be found on existing charts, is that taxiway labels are sometimes rotated to fit into taxiway geometry (e.g. “A-NORTH” in the bottom right image).

In consequence, precise rules had to be defined before the implementation of label generation for taxiway labels.

A first approach to mimic the current placement while optimizing the legibility could be to place the labels within the taxiway geometry if the taxiway is wide enough and next to it otherwise. However, this approach has two drawbacks. Taxiways are rarely wide enough to hold the complete labels with some margin to the edge of the taxiway even on maps of very large scale. Thus, the majority of labels would be placed next to taxiways. On the other side, this placement is not completely self-consistent as a fraction of labels would still be located differently.

Thus, the question remains if *all* taxiway labels shall be placed within or next to taxiways. The answer is already suggested by the official specification: label placement *next* to taxiways can be ambiguous if a label has similar distance to more than one taxiway.

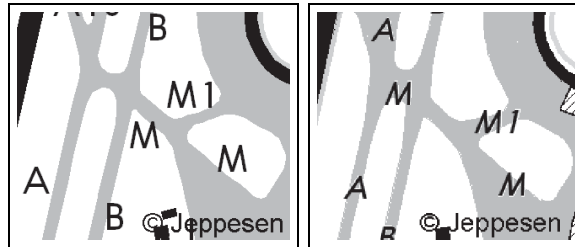


Figure 5.26: Taxiway labels placed next and within of taxiway geometries.

A typical case can be found in figure 5.26. The left image shows a current chart with labels placed next to the taxiways. Label “M” in the center cannot be associated definitely to either of the two intersecting taxiways at first view. Taxiway “B” has to be identified first by the labels on the top and the bottom before concluding that the remaining taxiway must be “M”. Similar arguments can be used for “M1”.

The right image shows the same situation, but now the labels are placed within the taxiways. Both taxiways “M” and “M1” can be identified instantaneously.

The faster information retrieval and unambiguity of labels placed within taxiways are a clear argument for this schema. These advantages cannot be outweighed by the slightly better legibility of labels placed next to taxiways due to their higher contrast.

The second observation from the analysis of existing charts was that some labels are rotated. Such a rotation is reasonable if the label exceeds a typical size and could cause conflicts with neighbor labels. Rotating such labels to fit within the taxiways is again a instrument to increase the unambiguity of a label, as the direction of the label coincides only with the direction of the own taxiway. To maintain legibility, the rotation angle has been limited to 60 degrees to both sides.

5.2.4.2 Implementation

According to the above specification, taxiway labels should be placed within taxiways. This has been implemented by guidelines representing the centerline of all taxiway elements belonging to one taxiway feature container. The centerline is computed as the polygon medial axis (“skeleton”) [Lee82, Aic95]. This procedure has been chosen over the use of the taxiline feature. The main taxiline is normally also located in the center of the taxiway element, but a large number of branch lines exist at intersections, which are hard to identify and remove reliably. Above all, the existence of a taxiline is not guaranteed at all. Thus, the use of the skeleton line is a step towards a more robust process.

If a taxiway feature container is composed of multiple parts, e.g. separated by an intersecting runway, a separate skeleton is computed for each fragment of the container.

Figure 5.27 illustrated the described guidelines (red) on the example of Berlin-Tegel. One

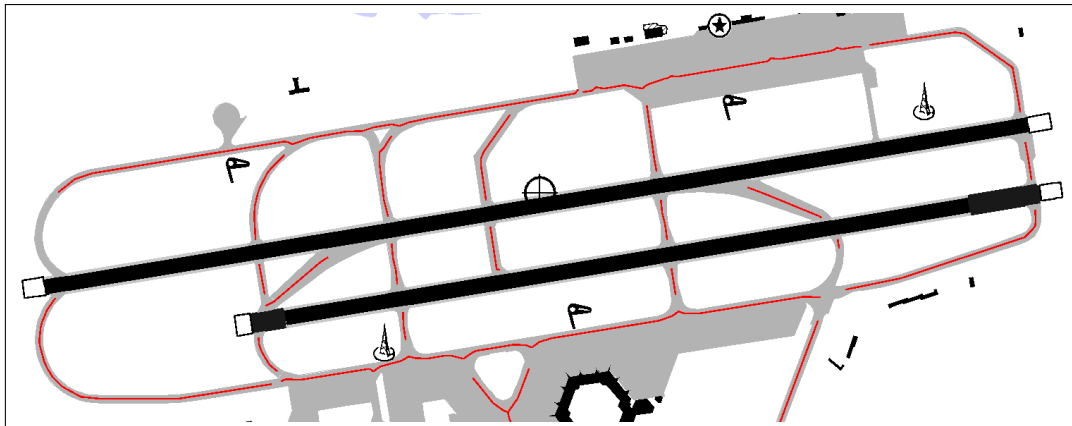


Figure 5.27: Guidelines for taxiway labels.

line stands for the allowed locations of candidates for one label. Very long taxiways are usually labeled by multiple labels. In such cases, the skeleton can be split into parts of the defined length (not shown).

In the current implementation, label candidates are placed along the guidelines at a distance of 50 meters, respectively $1/20$ th of the map width, whichever is the smaller distance.

Three different types of label candidates are generated:

- unrotated label candidates,
- rotated label candidates, if the length of the label text is equal or longer than 3 characters, and if the rotation angle is not more than 60 degrees, and
- line separated label candidates, if the text of the label contains whitespaces. This type is not rotated.

All types are generated simultaneously. It is left to the following candidate evaluation to determine, which type fits best.

The quality function for taxiway labels is defined by the following rules:

- Start value for each candidate location is 700.
- The overlap area of the label with the own taxiway is *added* to the quality value, such that full overlap results in 300 extra points. By doing so, the best matching label type (rotated/unrotated/multiline) can be identified. Combining the first two rules results in a maximum label quality of 1000, identical to those of other label types.
- Overlaps with other airport elements are subtracted in the known way. The only difference is that overlaps with runways are not generally forbidden, but rated with a high multiplier instead. This is done due to runway exits, which are sometimes too small to be labeled completely within their own outline. Applying these rules allows their labels to extend a short distance into the runway outline.
- Candidates located on intersections of two taxiways of different names are derated by 100 points. This is done in order to lower the chance that ambiguous candidates

are used in the labeling solution.

The other rules are already known from other label candidates: candidates close to the center of the guideline are preferred to those at their ends; candidates may not be located outside of the map area.

5.2.4.3 Results

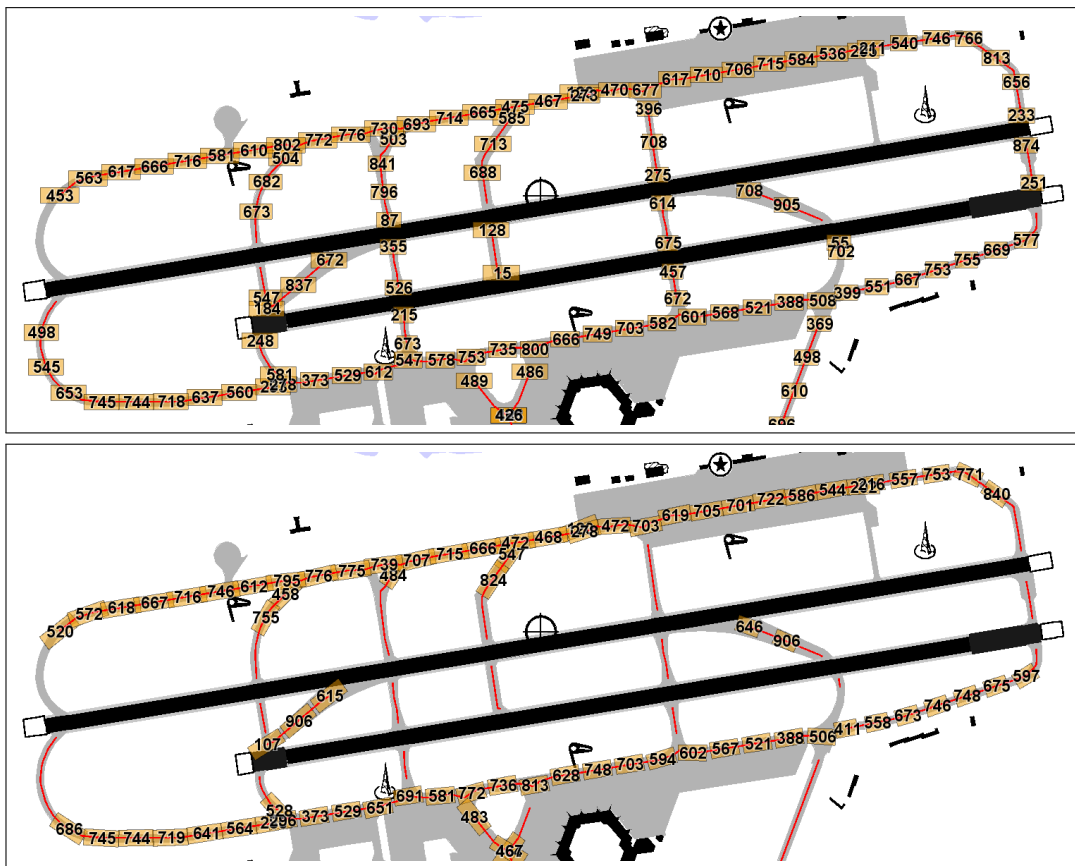


Figure 5.28: Candidates for taxiway labels.

Placing and evaluating candidates in the specified way along the guidelines results in the situation depicted in figure 5.28. For a better visibility, the upper image shows only unrotated candidates, and the lower one only rotated label candidates.

Comparing the two images reveals that rotated symbols are rated very similar to unrotated ones on taxiways running from east to west, like “NW” in the upper right corner. If the taxiway is oriented in other directions, the quality values differ significantly.

5.2.5 Parking Stands

5.2.5.1 Specification

The only formal rule for the placement of parking stand labels on today’s charts is that they should be charted as specified by source (AIP). As the AIP is compiled by individual airports, it can be assumed that each airport is using a different depiction, and that no specific, global rules are followed at all.

However, this is not very problematic, as the typical size of a parking stand is small compared to the average label size. In consequence, the placement of parking stand labels is dominated by the deconfliction component of adjacent parking stands. The exact location of a label relative to the stand area is only of secondary importance.

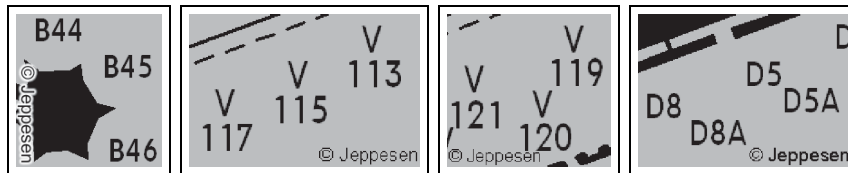


Figure 5.29: Placement of parking stand labels.

Nevertheless, a set of frequently used rules can be derived from existing maps (fig. 5.29). The default format of a parking stand label is a single-line label (left image). If the available space does not permit the placement of such labels, they can be hyphenated between leading character(s) and trailing digit(s) (second image from the left). This format is only possible with this naming schema. Hyphenations between leading digit(s) and trailing character(s) have not been found. If labels do still conflict, they are moved to alternate positions (third image). Parking stand labels are always unrotated.

Content wise, parking labels are frequently suppressed on the overview chart of large airports. Instead, a number of insets is created providing maps of different aprons at larger scale. Within these parking maps, parking stands are shown in full detail, including sub-stands like “D8” and “D8A” (right image in fig. 5.29).

The problem of identifying all elements belonging to such sub-stands has already been addressed in chapter 4.7.2. The required data are extracted from AMDBs in the common preprocessing phase and available as simple stand containers.

On charts of smaller airports, parking stand labels are placed on the overview chart together with all other types of labels.

5.2.5.2 Implementation

Due to the changed preconditions, the generation and evaluation of label candidates for parking stand labels differs from the algorithms used for other label types. As the typical size of a parking stand is of the same order as its label, a more systematic approach must be used to identify label candidates with a low risk of conflicts, respectively with a high

flexibility of locations which are still describing the parking stand good enough. This is done by defining preferred label candidates first.

A very good candidate location is usually the center of the stand area. This point has the lowest risk of conflicts with adjacent parking stands, while providing the most unambiguous labeling. A base quality of 1000 is assigned to this candidate. Exceptions to this rule are cases where two or more simple stand containers are collocated like in the “D8”/“D8A” example. As both stand containers are sharing the same stand area feature, the center for both containers would be identical and cannot be used here.

The second option for placing label candidates are published stand locations. Key advantage of stand locations is the combination of a unique identifier with a unique location as collocated stand locations like the stand area “D8_D8A” do not exist. Disadvantages of stand locations are that they are located at the very front of a parking stand, thus posing the candidate to the risk of overlapping a terminal building or other foreign map elements. Furthermore, all stand locations are usually located close together, thus providing only little flexibility for decluttering by using candidates at totally different locations.

Advantages and disadvantages of stand lines are exactly reversed. Stand lines are usually spanning the complete parking area along its major axis, thus providing a large variety of different candidate locations. On the other side, they may intersect with other stand lines, and hereby create the risk of ambiguous label placement in the intersection area.

Both stand location as well as stand line candidates are rated with a base quality of 900.

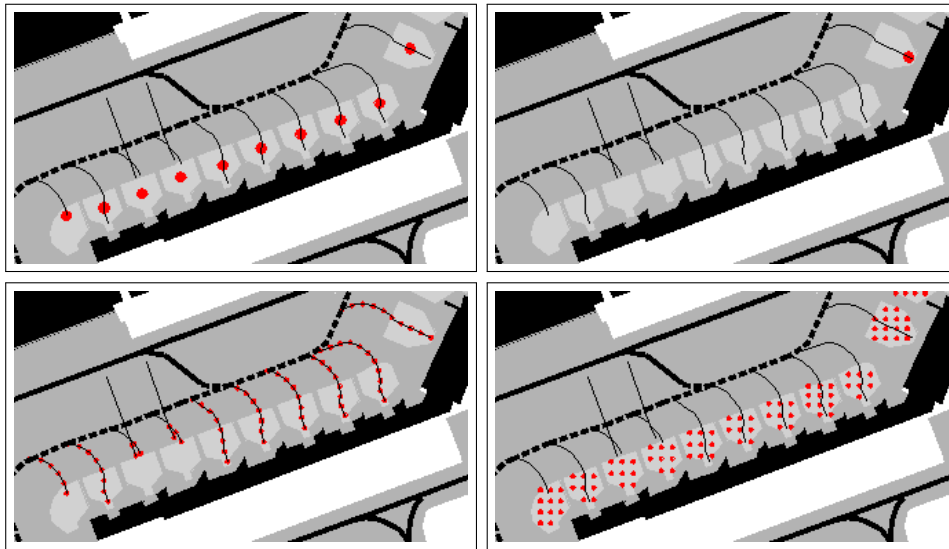


Figure 5.30: Candidate locations for parking stand labels.

As the existence of none of the preferred candidates is guaranteed, and in order to provide a larger set of candidates distributed equally across the complete stand area, additional label candidates must be generated. This is done by defining the stand area as guidance polygon and placing candidates on a raster within this area. Due to the small size of typical stand area, a label candidate is placed every 10 meters. A base quality of 500 is

assigned to these candidates.

The four defined candidate locations are illustrated in figure 5.30, starting with the center candidate location in the top left corner, followed by stand locations (only one instance), stand lines and finally the stand area raster.

If the parking stand identifier has a compatible structure, a set of additional, hyphenated label candidate will be placed at the same locations as regular label candidates. As multi-line candidates should only be used in extreme cases, a penalty of 300 points is assigned to each of these candidates.

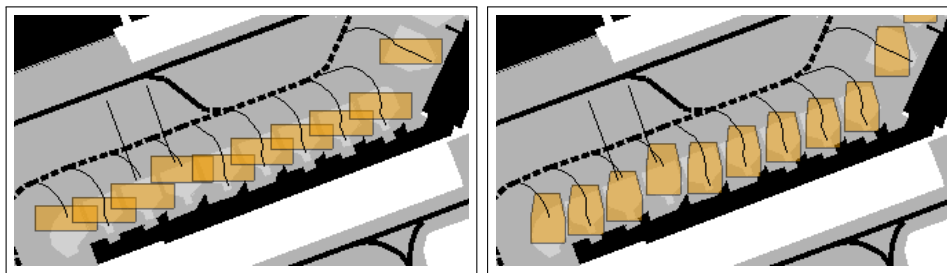


Figure 5.31: Candidates for parking stand labels.

Candidates for these two types of labels are shown in figure 5.31 on the example of parking stands “A24 - A42” at Terminal 1 in Frankfurt. Normal single-line label candidates, placed in the center of the parking stands, are shown on the left side. Nearly all of them are overlapping, so that a complete labeling would be impossible on this map scale. The right image shows label candidates for the same parking stands, but formatted in two lines. In this case, the candidates are not overlapping and can be placed without conflicts.

Using two types of label candidates thus increases the chance to be able to label all map features on maps of critically small map scales.

Label candidates are placed on all defined locations, but not shown for clarity as they are very packed on typical scales.

The following evaluation phase for parking stand label candidates is relatively simple. Based on the already assigned quality for the different locations, penalties are subtracted for two conditions:

- The distance of the label candidate to the center of the parking stand in meters times 4 is subtracted. This rule ensures that candidates are placed unambiguously near the center (if space permits). As the maximum size of a parking stand is about 80m x 80m, the maximum distance from its center can be approximately 57 meters and the max. penalty about 228. Thus, all candidate qualities are still valid after this step.
- Overlaps with other map elements are processed as in other cases before in order to optimize the legibility of the labels.

5.3 Label Placement

The final step in the labeling process is the placement of labels itself. Within this task, a set of label candidates is computed, such that the maximum number of map elements can be labeled without overlaps (*deconfliction*). This corresponds to the definition of the Label-Number Maximization Problem (LNMP).

The general approach taken matches the approach used during data preprocessing. First, data are reduced in a *deterministic* way. After that, *heuristic* methods are executed to come to the final labeling solution.

A large amount of third party research has already been conducted to solve the LNMP. This research was utilized here and adapted to honor label candidate quality and priority factors appropriately. The label placement module will be based on an algorithm proposed by Wagner et al. in their paper “Three Rules Suffice for Good Label Placement” [Wag01].

This algorithm is based on a conflict graph. In this graph, every label candidate is modeled as a node and every conflict with a candidate of *another* map element is modeled as an edge. The first phase of the core algorithm decimates the set of label candidates and places labels in a way which preserves the optimal labeling solution. The rules for this phase can be easily adapted to optimize also total feature priorities and candidate qualities. The implementation of the algorithm and the adaption to the present problem are described in section 5.3.1.

The second phase of the algorithm is heuristic and resolves conflicts not handled in phase I. The maximum number of placed labels, maximum priorities or maximum quality cannot be guaranteed anymore for data processed in this phase (see section 5.3.2).

5.3.1 Phase I: Rule-based Label Placement

The non-heuristic part of the deconfliction algorithm is based on only three rules which must be applied exhaustively. The rules introduced by Wagner et al. have been modified to consider the quality and priority factors of label candidates. Thus, not only the maximum number of placed labels is guaranteed, but the total quality and priority is maximized as well.

Rule L1: If a feature p has a label candidate p_i without any conflicts, declare p_i as part of the labeling solution, and delete all other candidates for p .

To optimize the total quality of the labeling solution, only candidates with a quality lower than p_i may be deleted, and p_i may only be added to the labeling solution if it is the label candidate with highest quality of this feature. Even if no label can be placed due to these restrictions, there is a chance that new, conflict-free candidates are generated due to the removal of lower quality candidates.

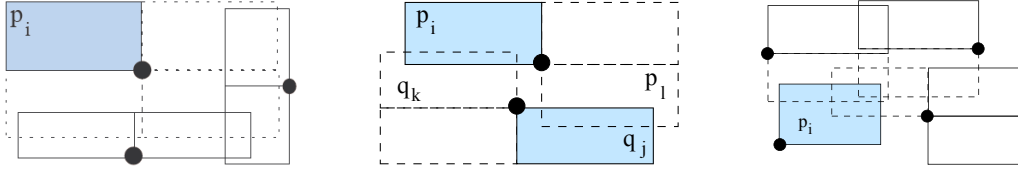


Figure 5.32: Deconfliction rules L1 to L3.

Rule L2: If a feature p has a candidate p_i which conflicts with some candidates q_k , and feature q has a candidate q_j ($j \neq k$) which conflicts only with p_l ($i \neq l$), then add p_i and q_j to the labeling solution and delete all other candidates of p and q .

Rule L2 has also been modified to optimize the quality of the labeling. In this case, only pairs of q_k and q_j are deleted whose combined quality is lower than the combined quality of p_i and q_j . Labels at p_i and q_j can only be placed if all conflicting candidate pairs could be deleted.

Rule L3: If a feature p has only one candidate p_i left, and the candidates overlapping p_i form a clique (overlapping each other), then declare p_i to be part of the solution and delete all candidates that overlap p_i .

An optimization of any properties of p is not possible, as p_i is already its last candidate. Instead, care has to be taken that no important candidates of other features are deleted. Therefore, candidates of other features are only removed, if the feature's priority is less than or equal to p 's priority.

Algorithmus 1 : Rule-based Label Placement Algorithm [Wag01].

Input : Label Features with unplaced Label Candidates

Output : Placed Labels

```

while One label has been placed do
  Apply Rule L1;
  if no Label placed by L1 then
    Apply Rule L2;
    if no Label placed by L2 then
      Apply Rule L3;
      if no Label placed by L3 then
        Apply Heuristic;
      end
    end
  end
end

```

To apply these rules exhaustively, the algorithm iterates through the rules as long as labels can be placed, including the prerequisite that unplaced label candidates are left. Pseudo

code of this iteration schema is shown in algorithm 1. While rule L1 can be applied to at least one candidate, only this rule is executed. Only if L1 cannot place a new label, L2 is applied and so on. If none of the rules L1 to L3 can place a new label, the algorithm executes the heuristic module, but jumps back to the rules after the first action of the module.

Due to the structure of the rules, a maximum of one label can be placed in L1 and L3, respectively two labels in L2. Restricting the rules L1 to L3 as done here for maximizing the total priority and quality of the solution reduces the number of labels placed successfully in the non-heuristic phase and leaves more work to the heuristic phase.

Wagner's approach has additional advantages coming in handy for the present problem. The rule-based algorithm represents the combinatorial part of the label placement problem, while the computation of the conflict graph is the geometrical component. Through the separation of these two parts, the algorithm can be easily adapted to different point-, line- and area labeling schemes by modifying only the construction of the conflict graph. The core algorithm is not affected.

It is even possible to handle non-rectangular label candidates. Thus, the label placement algorithm can be reused with minimal changes for the circular candidate shapes used in the airport moving map label generation system.

Furthermore, the rule-based algorithm runs pretty fast in $O(n + k^2)$ time, where n is the number of label candidates and k the number of conflicts. This is another key enabler for the use of this algorithm in the moving map system.

5.3.2 Phase II: Heuristic

The heuristic algorithm introduced by Wagner is relatively simple. It is based on the idea to remove troublemakers where the chance is great to find alternate candidates first. From the remaining features p the one having the maximum number of candidates left is chosen, and the candidate having the highest number of conflicts is deleted. This is repeated until each feature has at most one candidate left and no two candidates intersect. After the elimination of each candidate, the rules of phase I are applied again.

This means, that the heuristic does not place labels. It just removes conflicting label candidates. The difference to the rules is that this is done in a way which always terminates successfully, but which does not guarantee the maximum number of labels, quality, or priority any longer.

Even if it is not possible to guarantee the optimum priority or quality factors any longer, they can still be maximized. This has been done by modifying the heuristic such that the feature p having the maximum number of candidates and, if more than one with the same number exists, the one with the lowest feature priority is chosen. From this feature p , the candidate p_i having the highest number of conflicts and the lowest quality is eliminated.

By doing so, label candidates of higher quality, belonging to features of higher priority, persist longer. There is a chance that all their conflicts are removed sooner or later, and

the label can be placed in consecutive iterations of the phase I rules. Lower quality and lower priority features might not be labeled at all in such cases.

5.4 Results

This chapter presents and discusses airport diagrams generated with the developed systems. The main focus will be on the performance of the candidate generation and evaluation, the rule-based label placement, and the heuristic, but the basic maps will be mentioned as well.

The generation of an airport overview and a parking stand chart of Berlin-Tegel as depicted in figure 5.33 has been analyzed representative for medium size airports. Based on the common data processing, both maps are produced completely independent from each other.

For comparison, an additional combined overview and parking chart has been generated. Label placement on this map is much too dense, so that a large number of features cannot be labeled. However, studying the behavior of the algorithm in such extreme cases reveals more information than the analysis of standard situations does.

5.4.1 Quality of the Label Placement

Table 5.1 lists some statistic values collected during the label generation of these charts. Comparing the numbers for the two regular charts shows that in both cases more than 60% of the labels are eliminated and labels are placed by rule L1.

	Airport Overview	Parking Chart	Combined Chart
Features to be labeled	45	73	122
Labels Placed by L1	27	57	48
Labels Placed by L2	18	16	48
Labels Placed by L3	0	0	7
Unplaced Labels	0	0	19
Number of Candidates	837	1821	2905
Conflicts between Candidates	4251	12318	78740
Cand. Eliminated by L1	527	1354	871
Cand. Eliminated by L2	218	166	770
Cand. Eliminated by L3	0	0	23
Cand. Eliminated by Heuristic	47	228	1132
Iterations needed	86	296	1228

Table 5.1: Statistics of the label placement.

Even more interesting is the progress of candidate deletion and label placement (figures 5.34 and 5.35). The majority of label candidates are already deleted during the first three

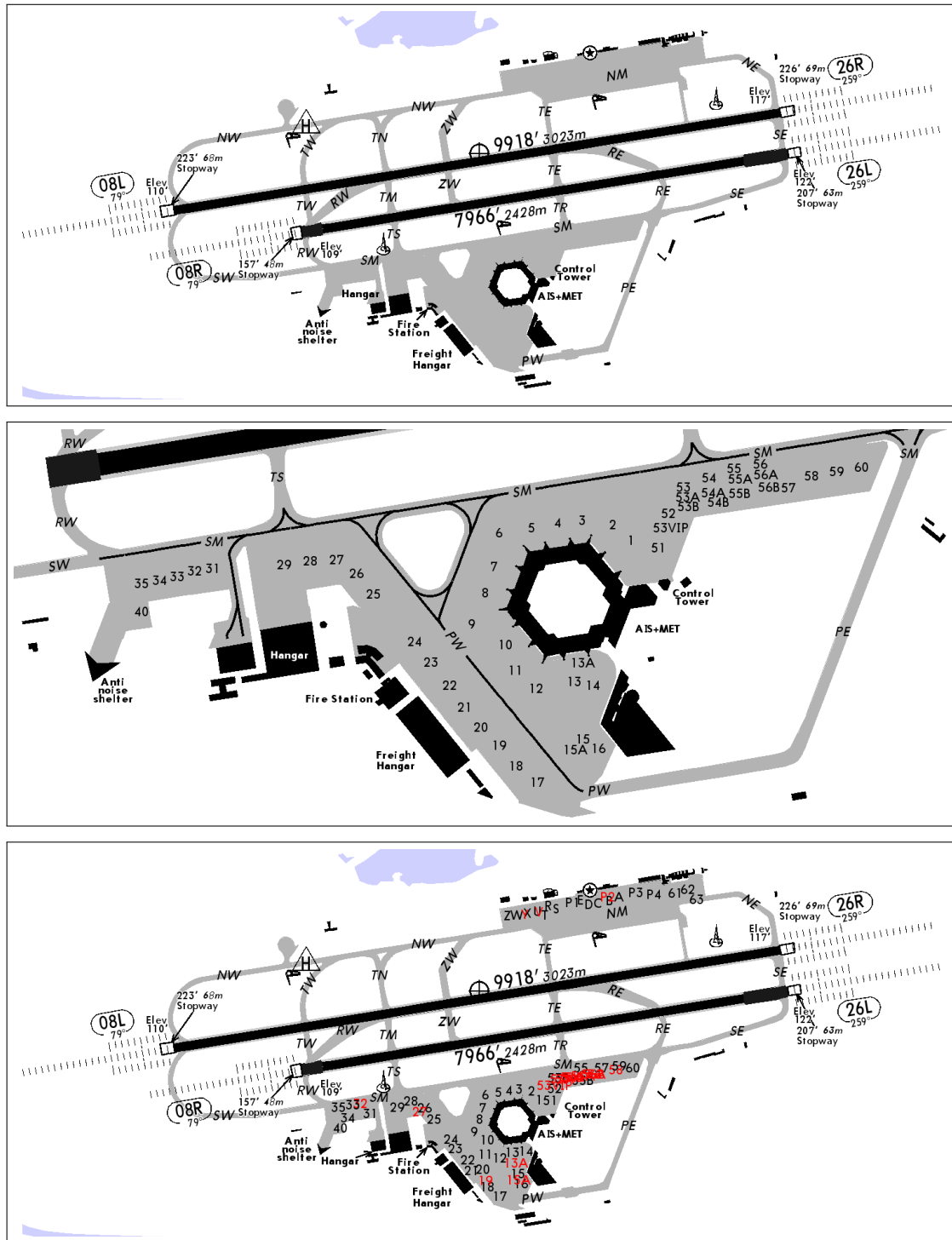


Figure 5.33: Automatically generated and labeled airport overview chart (top), parking chart (center) and combined overview/parking chart of Berlin-Tegel. Labels which could not be placed without conflicts are printed in red.

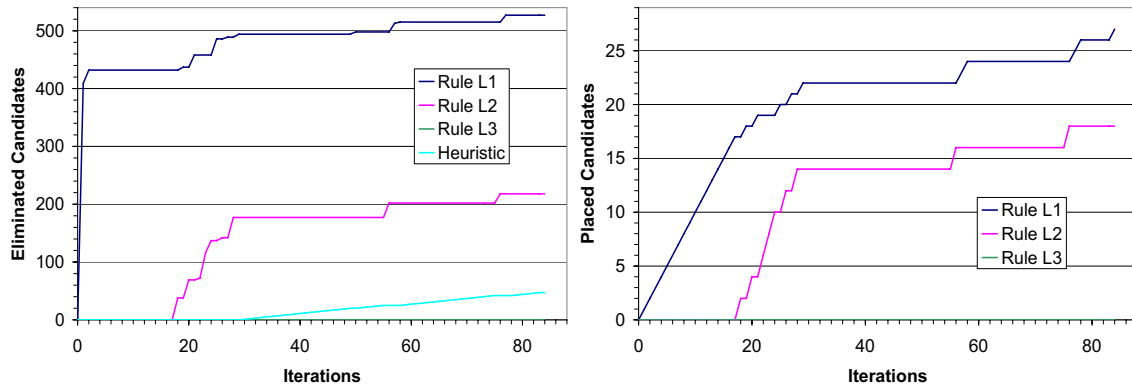


Figure 5.34: Performance of the label placement algorithm for the airport chart of Berlin-Tegel.

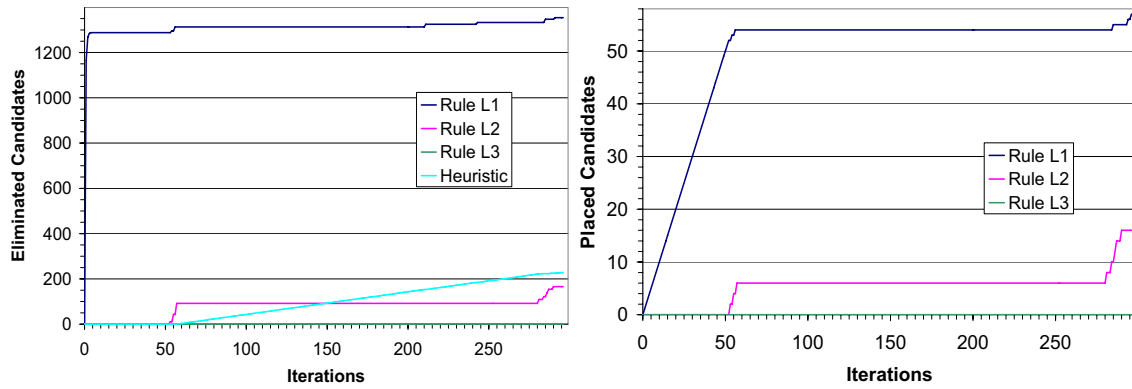


Figure 5.35: Performance of the label placement algorithm for the parking chart of Berlin-Tegel.

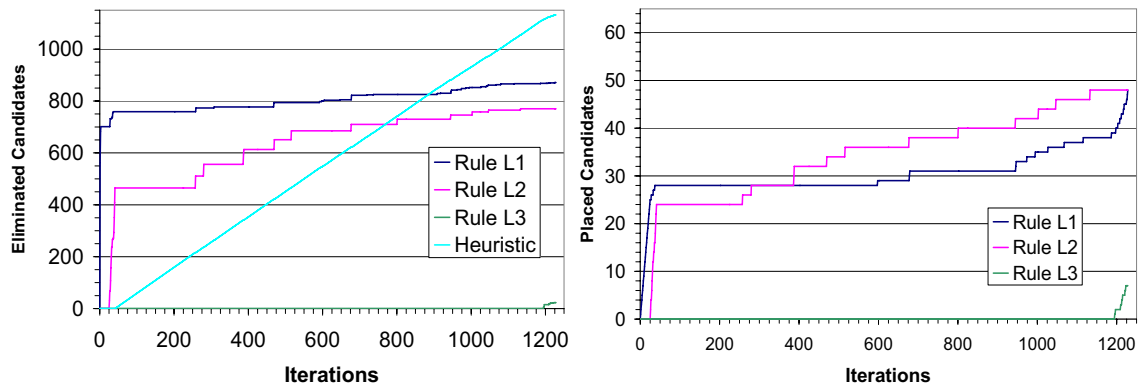


Figure 5.36: Performance of the label placement algorithm for the combined airport and parking chart of Berlin-Tegel.

iterations of the rules by rule L1. After that, the number of candidates remains constant for a while, while labels are placed by rule L1.

This process can be explained by the definition and implementation of rule L1. As stated above, the algorithm scans for features which do have at least one conflict-free label candidate, but due to the enhancement for quality maximization, other candidates of this feature are only deleted if they are of lower quality. However, this is done immediately. During the next iterations, exactly one label is placed by L1 in each cycle until no more conflict-free labels do exist.

Rule L2 is executed for the first time when no more labels can be placed by L1. The number of label candidates is already decimated enough in this phase, such that a number of features with only pairwise overlapping candidates exist. Labels for these features are placed, and due to the elimination of more candidates, L1 can also be activated successfully again.

Rule L3 could not be applied to any feature successfully. The restriction that all overlapping candidates must belong to a feature of equal or lower priority seems to be a large burden for this rule.

Finally, only complex clusters of conflicting label candidates belonging to multiple features are left. As rules L1 to L3 cannot be applied, the heuristic is activated and deletes candidates at a rate of one per cycle. This is done until a candidate cluster is simplified enough to enable the placement of the remaining candidates by rules L1 or L2. Removing candidates this way is slow compared to the rules L1 to L3, where all remaining candidates of a feature are removed at once when a label is placed.

All in all, only a small fraction of label candidates had to be deleted by the heuristic. All other labels could be placed in an optimized way. For the airport overview chart 35 out of the 45 labels have been placed at the best rated candidate location. Summing up the quality factors results in a very high total value of 34773 compared with a “perfect score”¹ of 36045. On the parking chart, 57 out of 73 labels could be placed at the best candidate location. The achieved total quality of 64412 from a maximum of 69659 is also very good.

Comparing the performance of the labeling algorithms on the separate overview and parking charts with its performance during the processing of the combined airport overview and parking chart reveals a similar behavior of the label placement system for the first iterations. About 50% of the features can be labeled by the rules L1 and L2 during the first 50 iterations (see figure 5.36). However, the system cannot complete the placement quickly as with the other charts because of a large number of clusters of overlapping label candidates. These must be removed one by one by the heuristics. This process requires about 1200 additional iterations.

As expected, the label placement system could not place all labels on this chart. 19 parking stand features are remaining unlabeled due to a lack of enough free space and conflicting

¹ The “perfect score” is computed as the sum of the best candidates of all features, regardless of any conflicts. This is not equal to the score of the optimum solution, which cannot be determined due to the NP-hardness of the problem.

other labels of higher priority.

The effect of the prioritization of parking stands as described in section 4.8 can be observed very nicely. Only parking stands having two neighbors of related designators are removed, as pilots can estimate their location from the context.

Only 52 of the 103 features could be labeled at the location of their best candidate, and 27 more on one of the locations of the second to the tenth best candidate. The total quality of all the labeling solution is 77325 out of a perfect value of 108960.

5.4.2 Runtime

The runtime of the system for the charts of Berlin-Tegel, grouped by major processing steps, is listed in table 5.2. The numbers have been measured on a standard laptop (IBM ThinkPad T60 with Intel Centrino CPU at 2GHz).

	Airport Overview	Parking Chart	Combined Chart
Data Loading	4.4s		4.2s
Preprocessing	6.2s		6.0s
Map Generation	2.6s	1.3s	2.6s
Candidate Generation and Evaluation	12.5s	6.1s	15.8s
Computation of Conflict Graph	3.7s	4.4s	28.0s
Label Placement	0.172s	1.52s	43.1s
Data Writer	5.6s		3.8s
Sum	49.0s		103.8s

Table 5.2: Runtimes of the label placement system.

Data loading and common preprocessing takes roughly 10 seconds. About 2.6 seconds are needed for the generation of the airport overview map, and half of this time for the smaller parking map.

The generation and evaluation of label candidates for the separate overview and parking charts lasts about 12.5 seconds respectively 6.1 seconds. This is the longest runtime of any module for these two charts and can be explained by the large number of on average 19 respectively 24 label candidates per feature, which are generated and evaluated using costly geospatial operations. The computation of the conflict graph uses also spatial operations, but most of them have been optimized by the use of a geospatial index.

The final label deconfliction runs amazingly fast and takes only 172ms for the airport overview map and 1.52s for the parking map. From these times, only 125ms are needed for the first iterations utilizing the rules L1 to L3 exclusively. The according value is 234ms for the parking map. The remaining time is needed for eliminating label candidates by the heuristic.

This trend is even clearer for the combined overview/parking chart. In this case about 2.3s are spent placing 50% of all labels by rules L1 to L3 while additional 40.8 seconds are

required by the heuristic to eliminate more candidates and pave the way for placing the remaining labels.

This effect is further emphasized by the optimization of the rule-based system for quality and priority. Through these restrictions, deconflicted but suboptimal located labels cannot be placed by the rules L1 to L3 anymore. Instead, existing conflicts of higher rated candidates of the same feature must be resolved by the heuristic first.

5.4.3 Quality and Legibility of the generated Maps

The automatically generated airport diagrams presented in figures 5.33 and appendix A have a very good legibility. All labels are placed unambiguously and can be clearly associated with their map elements.

The depiction of the map elements and the labeling is close to current paper charts. Existing differences can often be attributed to inconsistencies in the existing charts. An example are taxilines. Usually taxilines are depicted on taxiways connected to aprons and are hidden otherwise. However, the existing parking chart of Kansas City in figure A.1 on page 171 does not show taxilines and leaves the pilots in uncertainty where the taxiways are located exactly. The automatically generated chart visualizes this scenario precisely and consistently.

Another source of discrepancies are different effective dates of the data. The automatically generated maps of Kansas City in figures A.2 and A.1 are showing a number of temporary construction areas around the terminals. This construction zones are not displayed on the paper chart.

However, some flaws can also be found on the automatically generated charts. On figure A.2, Terminals B and C are labeled on the apron-side of the airport instead of the ground-side. According to a general cartographic rule, labels should be placed in the area to which their feature belongs logically [Imh62a]. As terminal buildings belong to the ground-side of an airport, their labels should also be placed there as done with the label “Terminal A”. Such a requirement does not exist in today’s charting rules and has thus not been implemented.

Taxiway broken into multiple, short segments by intersecting other taxiways, like taxiway “K” on the same map, are currently labeled per segment. While this solution is perfectly correct and clear, it incorporates a certain risk for cluttering.

Another taint can be observed in the charts of Berlin-Tegel in fig. 5.33. While most taxiway labels have been placed unrotated, the two labels for “RW” and “RE” have been rotated to fit completely into their associated taxiways. While this is consistent with the postulated placement rules and still legible, it generates a slightly unsettled appearance.

5.5 Summary

The depiction of map elements, symbology, and the labeling on the automatically generated airport maps is close to current paper charts. Differences can often be attributed to inconsistencies in the existing charts. Especially exact outlines of buildings and aprons differ quite frequently.

The legibility of the automatically generated labeling solutions is very good. The labels are placed unambiguously and can be clearly associated with their map elements. This proves that the developed label candidate generation models and the related candidate evaluation functions are working properly.

The runtime of the complete algorithm is acceptable with about 30 seconds per generated chart. The largest portion of the runtime is required for the label candidate generation, evaluation, and the construction of the conflict graph on the one side, and for the heuristic placement on the other side.

The rule-based label placement has shown great potential. It is very fast while computing labeling solutions of large size and high quality.

Modifying rule L1 to place more than one label per iteration, and the heuristic to delete more than one candidate has a good potential to speed up the runtime significantly for maps with dense label constellations.

Chapter 6

Label Generation for Airport Moving Maps

The concepts and algorithms used for the label placement of static airport maps have been presented in the previous chapter. This chapter will describe the steps needed to refine this work for labeling dynamic airport moving maps.

Special considerations to be taken into account for airport moving map systems are:

- Resolution and contrast of today’s electronic displays are much lower than those of paper prints. In order to maintain legibility, labels must be printed in a larger size. In consequence, fewer items can be labeled without conflicts on a map of same scale. The prioritization of features becomes more important to separate important from unimportant map elements and label only the important ones.
- Electronic maps can be rotated into arbitrary directions. Labels must be deconflicted for all map orientations.
- Electronic map displays offer different zoom ranges. Labels must be deconflicted for all zoom levels. A label should be placed at the same location on all zoom levels to avoid unsteady effects when switching zoom levels.

The general architecture of airport moving map systems is predetermined by strict certification requirements for airborne software [DO178B]. Due to the costs of such a certification, a general design principle in avionics is to perform only absolutely necessary processing steps onboard the aircraft. All other data processing is performed on the ground and precomputed results are loaded onto the avionics systems.

The labeling concept outlined in the ARINC 816 specification implements this design in a very simple way: only one label candidate per map feature is generated (“anchor point”) and stored in the dataset. The location of this anchor point is determined by geometric properties of AMDB data, but it is not deconflicted! Priorities of anchor points are not included. Without doing a complete label placement with candidate generation and evaluation on the airborne side, the options are limited to either place all labels regardless of conflicts or to place only some and delete all others based on an ‘arbitrary’ deletion

schedule [Psc07]. A normal deconfliction is not possible in the AMM as no alternate candidate locations are available.

The Jeppesen Airport Moving Map, which is used to demonstrate the capabilities of the new label placement system, follows also the architecture to generate label positions offline. However, in this case a deconflicted labeling solution can be computed offline for a discrete number of zoom ranges through a closer coordination of the offline label preprocessor and the online moving map component. The resulting datasets are added to the aerodrome mapping data and loaded into the moving map application onboard the aircraft (fig. 6.1).

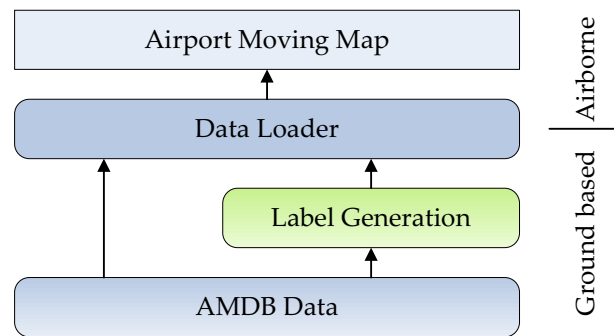


Figure 6.1: Overview of the label generation system for the electronic moving map.

The following principles have been applied to realize the system within the given environment:

- Different label types and different levels of detail are displayed on the various zoom levels. The hierarchical data structures generated in the common preprocessing phase will be used as data source.
- Labels are not rotated with the underlying map, but are always rendered upright.
- To ensure that a label candidate does not conflict with another candidate in an arbitrary map orientation the circumcircle around the label candidate is used for the conflict determination.
- To minimize the size of the circular label candidates, the label text is hyphenated where possible.
- A labeling solution will be computed for the regular AMM zoom levels of 'all airport', 2.0nm, 1.0nm, and 0.5nm. In addition, the application has been extended for a fifth zoom range of 0.25nm. This is mandatory to display complex parking stands such as the "D10"/"D10A" example in Frankfurt in full details and also supports a good display of aprons with very small and dense parking stands like the general aviation ramp in the south of Frankfurt/Main.
- Extended map elements like taxiways must be labeled at regular distances, such that at least two labels are visible for any field-of-view.
- If possible, labels should be placed at the same location on all zoom ranges to avoid unsteady effects when switching zoom levels.

- If map features cannot be deconflicted completely, the feature label with the lower priority is not displayed. A warning is posted to the operator.
- Overprints of map elements by a label are only rated to avoid heavily ambiguous candidate locations. The contrast between the label text and the background map is hard to evaluate offline as it changes potentially with map rotation. Instead, a halo around the font is used to guarantee a good legibility on all map backgrounds.

6.1 Generation and Evaluation of Label Candidates

6.1.1 General

A moving map display has a fixed number of pre-defined zoom ranges. Different zoom levels are used to display different amounts of data in an optimized way on the small electronic display. Table 6.1 provides an overview of the existing zoom levels, and the label types to be placed on each of them.

	All Airport	2.0nm	1.0nm	0.5nm	0.25nm
Runways	Yes	Yes	Yes	Yes	Yes
Buildings	Yes (major only)	Yes	Yes	Yes	Yes
Taxiways	No	Yes	Yes	Yes	Yes
Parking	No	No	Yes	Yes (Com-	Yes (Simple
Stands			(Parking Containers)	plex Stand Containers)	Stand Containers)

Table 6.1: Labels to be displayed at various zoom ranges on the airport moving map.

6.1.2 Runways

Runways are the most important elements on airports. They must be labeled at all zoom levels. The runway identifier label is placed at the beginning of the respective runway or, if this point is outside of the visible area, at the position where the runway intersects the edge of the display. The label is rotated perpendicular to the runway direction. As the map itself is rotating too, the label is displayed upright when an aircraft is lined up on this runway. Due to the importance of the runway identifier label, this functionality is already implemented directly into the airport moving map. A separate label generation is not required. Other runway labels like the runway length or the runway elevation label found on paper charts are not displayed on the moving map. This information is creating additional clutter on the map, while not supporting the situational awareness of the aircrew. The

operational concept of a moving map – respectively of an electronic flight bag – is that this information can be retrieved from other applications such as approach charts if required.

6.1.3 Helipads

This moving map is intended only for use in fixed wing aircraft. The label and the triangle symbol for helipads are of no relevancy to the users. Thus, helipads will not be labeled to avoid this type of cluttering.

6.1.4 Buildings

6.1.4.1 Specification

Large buildings like passenger terminals or large freight hangars are frequently used for the first orientation on airports. Accordingly, they must be labeled on all zoom levels of the airport moving map, including the largest 'all airport' view. Labels for smaller buildings are added on higher zoom levels.

Labels are placed on top of their respective buildings or such that they at least overlap them.

6.1.4.2 Implementation

This placement concept as been implemented similarly to the label candidate generation for static airport maps. First, a guidance geometry enclosing all possible candidate locations is defined. This geometry is based on the convex hull of the outline of the building. The convex hull is used to allow also label candidates in patios or concave corners of the building. If for some reason another label can be placed there as well, the label deconfliction has to resolve this conflict via the priorities of the features and alternate label candidates. This guidance polygon is further on extended by the minimum of label height and label width to support labels which are not located right on top of the feature, but are overlapping it.

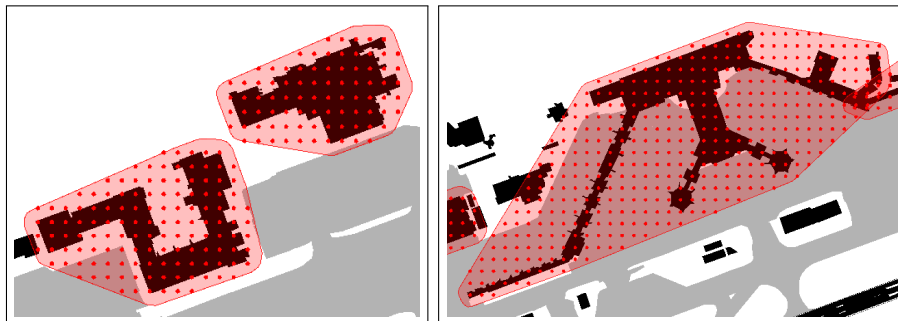


Figure 6.2: Guidance polygons for AMM building labels.

Two examples of the resulting guidance polygons are presented as red shared areas in

figure 6.2 for two cargo hangars and Terminal 1 in Frankfurt. These figures also show the candidate locations, placed on a raster of 50 meters, as small red dots.

The evaluation of label candidates is also done using the tool set introduced in chapter 5. An initial quality of 700 is assigned to all label candidates. Overlaps with the own building are rated positive with a maximum of 300 additional points, so that label candidates located completely within their own building have a quality of 1000. This results in very favorable label locations, as the assignability of such labels to its feature is maximized and conflicts with other labels are very unlikely.

Overlaps of a label candidate with other buildings are rated negative with a maximum penalty of 1000 assigned to candidates overlapping other buildings completely. This effectively removes the candidate from the list of usable label locations.

Overlaps with aprons or parking areas are also rated negatively, as overlaps with parking label candidate are very likely. Degrading such candidates results in candidates preferably being placed directly on the building or on the land side of the airport, where no other relevant map features do exist.

Finally, a maximum penalty of 200 points is assigned for the distance from the building center to the label candidate. The intention behind this rule is to promote candidates placed in the center of their buildings.

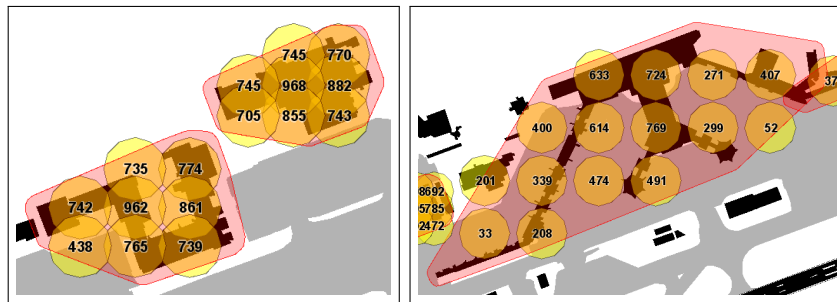


Figure 6.3: Evaluated label candidates for AMM buildings.

Figure 6.3 illustrates the generated candidates and assigned qualities. The label candidates are sketched as orange circles. The circular shape ensures that a candidate does not conflict with any other candidate while the map rotates. The best rated label candidate is located close to the center and overlapping the building in all cases. For clarity, candidates are spaced at a distance of 200 meters instead of 50 meters in both screenshots.

6.1.4.3 Layer Structure

As mentioned in one of the previous sections, the different zoom ranges of the moving map are used to display various levels of details. The following concept has been implemented for buildings.

On the 'all airport' scale, only buildings of operational importance like Towers or Fire stations, and the largest terminal and freight hangars are shown. The actual determination of "largest" is left over to the deconfliction algorithm. Each building is assigned a priority.

The maximum value of 1000 is used for towers and fire stations. The priority of all other buildings is the sum of a base quality of 500 plus a maximum of 500 additional points depending on the fraction of own building area by largest building area. During label deconfliction, the most important labels are placed first, which is effectively a placement by their size. The normal label candidate generation and evaluation as described above is applied.

From zoom level 2.0nm up to zoom level 0.25nm all buildings are labeled. If the building has not already been labeled on the next smaller zoom level, the standard algorithm for the 'all airport' level is used to compute label candidates and priorities.

Otherwise, the priority and the location is copied from the label at the next smaller zoom level. In general, this location is already decluttered and no further steps are mandatory. However, as more map space is available at this level, there might be the option to move the label to a better location. On the other side, new labels can be activated at this zoom level, causing new conflicts. Therefore, the best candidate from the next smaller zoom level is tested at this level again to provide an adequate amount of flexibility for both cases.

The concept of operation for airport moving maps assumes that higher zoom levels of airport moving map displays are primarily used for monitoring taxiing and approaching the gate. Thus, taxiway and parking labels become more important. In consequence, the priority of building labels is decreased by 500 units from zoom level 0.5nm on. In case of a conflict, the building label will be removed in favor of the other label.

6.1.4.4 Results

A series of building labels at various zoom levels of the airport moving map is depicted in appendix B.1 and B.2 on page 184. The layering concept works pretty well. Only significant buildings are labeled at the two presented airports on the 'all airport' view, while labels for additional buildings are added on higher zoom level if space permits. All labels are placed nicely within their buildings.

6.1.5 Taxiways

6.1.5.1 Specification

Taxiways are important airport elements for pilots when taxiing. Thus, they must be labeled with high priority at zoom ranges typically used in this phase of flight, which are the 2.0nm and 1.0nm zoom levels. Taxiway labels are less important at the higher zoom levels of 0.5nm and 0.25nm as those are usually used for approaching the gate and docking. However, taxiway labels are still shown in these views if space permits.

In contrast to buildings or parking stands, taxiways frequently have a long stretched, linear shape. Thus, it is not appropriate to label them with just one label. Instead, a discrete number of labels have to be placed along the structure of the taxiway, such that

the taxiway can be identified unambiguously in all orientations and fields-of-view.

6.1.5.2 Implementation

Taxiway containers are used as data source for the label placement of taxiways. The skeleton computed from the taxiway outline is used as guidance geometry similar to the implementation for static airport charts.

The guidance line is split into multiple line segments in order to support multiple labels per taxiways, whereas each segment will be labeled separately. The maximum length of each segment will be 45% of the screen width. Thereby, it can be guaranteed that at least two labels per (long) taxiway are visible within the current field-of-view.

Label candidates are placed along the guidance line (segments) at distances of 50 meters or 1/20th of the screen width between each other, whichever is smaller. The resulting dataset is similar to the one presented in figure 5.28 on page 143. The only difference is the shape of the candidates, which is now circular to reflect the rotation of the label in front of the map background.

Rules for the evaluation of taxiway label candidates are also derived from the solution for static maps:

- The start quality for candidates is 700.
- Overlaps with the own taxiway segment are added to the quality with a maximum of 300 additional points for a full overlap. Thus, the maximum quality for a taxiway label candidate is 1000 and equal to the quality factor for other candidate types.
- Overlaps with other airport elements are subtracted from the quality factor. Care is especially taken on overlaps with other taxiways, as those are highly likely in the zoom levels of 1.0nm and 0.5nm. Intersections with runways are generally allowed to provide some flexibility for labeling runway exits, but are rated with a high penalty factory.
- Last but not least, label candidates located on intersections between two taxiways are derated by 100 points as those positions are ambiguous to some extent.

These rules are applied to all generated label candidates. Candidates with a quality of equal or less than zero are not considered in the label placement.

6.1.5.3 Results

Taxiway labels generated using the above label candidates and candidate qualities are displayed in figures B.7 to B.10 in appendix B.3 on page 190.

Most taxiways can already be labeled without conflicts at the 2.0nm zoom level, though the result looks packed for some areas of the airport. At 1.0nm, the labels look less packed due to the higher zoom scale, and enough space is available to place also larger labels like “A-NORTH”. The concept to place two to three labels for each field-of-view works very

well. Those taxiways can be clearly identified and their direction becomes visible as well (e.g. taxiways “C” and “G”). Both can be done without adding clutter to the display.

6.1.6 Parking Stands

6.1.6.1 Specification

Parking stands can only be labeled reasonably at higher zoom ranges because their large number would cause too many conflict on lower scales. Therefore, three levels-of-detail have been implemented:

At the highest zoom level (0.25nm), all details of a parking stand will be displayed. This includes different labels for sub-elements of the parking stand like the stand guidance lines “D10” and “D10A” in Frankfurt.

At the next lower zoom level (0.5nm), only the main parking element “D10” will be shown. To minimize cluttering, it is even possible to suppress some of these labels as long as they can be reconstructed mentally by the users. This is done by lowering the priority of the intermediate features (see sec. 4.8).

Only groups of parking stands (e.g. “D1 - D13”) will be labeled on the 1.0nm zoom scale.

No parking stand designators are shown on lower zoom levels.

6.1.6.2 Implementation

The three levels-of-detail specified above correspond to the simple stand container, complex stand container, and parking container data structures generated by the common preprocessing steps. Accordingly, these data will be used as input for the label candidate generation at each of the three zoom levels.

Label candidates for the 1.0nm zoom level are constructed from the parking container structures. A label candidate is placed at the center of each contained stand container. The quality is computed by subtracting the distance of the candidates to the center of the parking container from the initial quality of 1000. This results in preferred locations close to the center of the parking container.

The generation of label candidates for the 0.5nm and 0.25nm zoom levels from simple and complex stand containers has been taken over from the process for paper charts as described in section 5.2.5 in detail. First, a number of label candidates are generated for special locations of a parking stand. These are the stand locations, along the stand lines, and in the center of the stand area. After that, additional candidates are generated on a discrete raster within the stand area with a distance of 10 meter between the coordinates.

The base qualities for the candidates are assigned depending on their origination. The candidate located at the stand area center has a quality of 1000, stand location and stand line candidates 900, and all others have a base quality of 500. The further candidate evaluation is based on the following rules:

- Overlaps with other airport elements are reducing the quality of a label. Especially

buildings, taxiway, and aprons are considered here in order to prefer label with a low risk of conflicts.

- The distance from the center of the stand area is also subtracted from the quality. This is also done to prefer unambiguous labels near the center of the stand area.

6.1.6.3 Results

Examples of parking labels for all three zoom levels are depicted in appendix B.4 on page 194. The level-of-detail concept works very well. The groups of stand areas labeled at the 1.0nm range provide a first input to the pilot where to look for a particular parking stand without adding too much clutter to the display. The following zoom levels of 0.5nm and 0.25nm provide gradually more information if required by the pilots.

Nearly all labels can be placed at 0.5nm, but labels are placed very dense for groups of small parking stands like “V119 - V130”. The result is an unsteady layout and single unlabeled features. Thus, a higher zoom level like the 0.25m range implemented in this demonstrator is advisable.

6.2 Label Placement

The rule-based label placement process introduced for static airport maps in chapter 5.3 on page 147 is used for placing labels in the moving map system as well.

First, a conflict graph is computed from the set of all label candidates. Based on this graph, phase I of the placement algorithm is responsible of reducing the set of candidates and place labels in a deterministic way by following the three rules L1 to L3. During this phase, the maximum number of labels, the maximum overall priority, and the maximum quality are maintained.

Second, the heuristic phase processes conflicts which cannot be resolved with the optimizations of phase I. The algorithm iterates through both phases until all labels are placed or until all unplaced label features do not have label candidates left.

6.3 Results

Table 6.2 shows a number of statistic values collected during processing of Frankfurt/Main for the AMM system.

In general, a relatively low number of conflicts between label candidates is observed compared with the data of the static airport maps (table 5.1). This is a first indication that the selection of airport features to be displayed on the various zoom levels is reasonable.

An exception is the 0.5nm zoom level, where the number of conflicts relative to the number of candidates is significant higher than for other zoom levels. More iterations are required for placing labels at this zoom level. Both are indications that the labels are very dense at this zoom level.

	All Airport	2.0nm	1.0nm	0.5nm	0.25nm
Features to be labeled	8	112	184	428	494
Labels Placed by L1	4	78	149	337	412
Labels Placed by L2	2	10	20	58	62
Labels Placed by L3	1	13	7	13	10
Unplaced Labels	1	11	8	20	10
Number of Candidates	205	936	1289	6052	3296
Conflicts between Candidates	1488	4481	2834	23888	7842
Cand. Eliminated by L1	81	541	872	4197	1793
Cand. Eliminated by L2	58	91	110	777	787
Cand. Eliminated by L3	1	19	14	22	12
Cand. Eliminated by Heuristic	58	184	116	643	217
Iterations needed	64	285	285	1034	675
Total Quality	4964	34653	81166	250505	288809
Perfect Quality	5547	38720	88876	277578	303139

Table 6.2: Statistics of the label placement.

The 0.25nm zoom scale is less critical as much more space is available. The 1.0nm level on the other side is uncritical as well as much less features are to be labeled. The only difference between the 0.5nm and the 1.0nm range are the added parking stand labels. Thus, they must be the reason for this effect.

A similar conclusion has already been drawn in section 6.1.6.3 where the unsteady labeling of dense parking stand groups has been identified.

In consequence, care should be taken when processing data at this zoom level. Mitigations would be to either change the scale slightly (e.g. to 0.4nm), or to add more and/or stricter decluttering rules for parking stands.

As observed before, the majority of labels are placed in phase I of the label placement algorithm. The results are high quality labeling solutions, reaching typically 90% to 95% of the perfect quality score ¹.

The runtime of the label placement system, grouped by it's major modules, is listed in table 6.3. The majority of the processing time is related to the generation and evaluation of label candidates and the computation of the conflict graph. A part of this time can be saved by optimizing the label generation further to produce fewer candidates. However, this does not influence the validity of the concept as these modules are only used in an offline labeling solution.

Even porting the system to a completely dynamic online labeling solution is within one's

¹Perfect quality is defined as the sum of the best candidates of all features, regardless of any conflicts. This is not equal to the score of the optimum solution, which cannot be determined due to the NP-hardness of the problem.

	All Airport	2.0nm	1.0nm	0.5nm	0.25nm
Data Loading			5.13s		
Preprocessing			32.02s		
Candidate Generation and Evaluation	9.59s	29.91s	33.89s	55.52s	75.20s
Computation of Conflict Graph	2.45s	6.44s	4.97s	37.34s	12.17s
Label Placement	0.109s	0.953s	1.094s	10.141s	2.562s
Corrected Label Placement	0.109s	0.37s	0.1s	0.24s	0.02s
Data Writer			2.9s		
Sum			323.0s		

Table 6.3: Runtimes of the label placement system.

grasp by using this system. All preprocessing, including the candidate generation, evaluation, and the computation of the conflict graph can be done offline and the results loaded into the online system. Thus, the timing of these components is not critical. The online system has to perform only the label placement itself. The above numbers are the times required for labeling the complete airport for the specified zoom level. An online system would only have to label the current field-of-view, reducing the above runtimes. According times are listed as “Corrected Label Placement”. These times are shorter than one second, thus enabling a constant update of the labeling solution with an acceptable update rate. Further optimization, including the use of C/C++ instead of Java, the optimizations of rules L1 and the heuristic as outlined in section 5.5, and mechanisms to update the labeling solution only in changed areas can further improve these runtimes.

6.4 Summary

The generated labeling solutions have been manually inspected for a number of airports and the results have been found to be very good. All placed labels are fully deconflicted at all zoom levels and orientations and can be easily and unambiguously associated with their map elements.

About 90% to 95% of map elements are labeled automatically, and only a small fraction cannot be labeled due to a large number of conflicts. Some of them are not part of the optimum solution and cannot be placed at all, even by manual rework, but exact numbers cannot be computed due to the NP-hardness of the problem.

The degree of information displayed at various zoom ranges is very reasonable and supports the pilot’s awareness of the surrounding airport.

The hierarchical data structures generated during the common preprocessing of the airport data have proven their potential. Labeling Buildings, Taxiways, and Parking Stands with various levels-of-details at different zoom ranges is easily possible based on these structures. The zoom level of 0.5nm needs a redefinition of features to be displayed (in particular

parking stands) or a change in the scale (e.g. to 0.4nm). An additional zoom level of about 0.25nm is advisable for a fully deconflicted display of parking stands in full details.

Chapter 7

Conclusion and Outlook

The development of the label generation system for airport diagrams showed that a thorough data aggregation and decluttering are the foundation for a good solution.

According data structures to store aggregated airport data in a hierarchical tree have been introduced and algorithms have been developed to populate these structures. A major problem towards reaching this goal is that the layout of existing airports and used designators are following no standards. Hence, a number of heuristic filtering and validation steps must be applied to resolve related database artifacts and to achieve consistent and reliable results.

Based on the hierarchical data structures, display systems or chart renderers can perform the actual label placement with a set of information usable for a particular view. The potential of the new datasets has been demonstrated on two applications.

Static airport maps have been generated using styles and parameters from their existing, manually composed counterparts. All automatically generated airport diagrams have a very good legibility. The labels are placed unambiguously and can be clearly associated with their map elements. The depiction of the map elements and the labeling is close to current paper charts, whereas differences are frequently caused by inconsistencies on existing charts.

Second, labeling solutions for a dynamic airport moving map display have been computed. The placement of the labels is also very clear and unambiguously. Various levels of information are displayed at different zoom ranges, thus supporting an optimized information retrieval by pilots and clearly demonstrating the potential of the hierarchical data structures.

It became obvious during the tests of this application that the highest zoom level of 0.5nm is not sufficient for placing all labels at the maximum level-of-detail. The addition of another zoom level of 0.25nm is advisable.

The label deconfliction mechanism used in both demonstrators is based on an existing, rule-based algorithm which has been modified to consider feature priorities and label candidate qualities. A high fraction of labels is placed automatically, and the runtime is

very good as long as the number of label conflicts is limited. Approaches to optimize the runtime have been identified and pointed out.

The business rules used for the label generation for static airport maps as well as for the airport moving map application have been derived from existing solutions. While those products are based on years of cartographic experience, a human factors evaluation could outline room for further enhancements.

The presented label placement system is the first step towards an automatic system to generate airport maps respectively databases for airport moving map applications. In order to set up a production system, the labeling component must be embedded into a complete process chain. On the input side, a change management process must monitor the source databases and kick off the production of a new map or AMM dataset when relevant information has changed. On the output side, tools for verification and manual edits are required. Finally, data delivery tools and data links are required to deliver the data to the users and into the cockpit in real-time.